

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт электронного обучения

Специальность программное обеспечение вычислительной техники и автоматизированных систем (230105)

Кафедра автоматики и компьютерных систем

ДИПЛОМНЫЙ ПРОЕКТ/РАБОТА

Тема работы
Информационная система учета элементной базы лаборатории студенческого научного творчества

УДК 004.422.63:378.147.88:001.89

Студент

Группа	ФИО	Подпись	Дата
3-8001	Савельев Николай Михайлович		

Руководитель

Должность	ФИО	Учебная степень, звание	Подпись	Дата
Ассистент	Мартынов Я.А.			

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Учебная степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Учебная степень, звание	Подпись	Дата
Ассистент	Невский Е.С.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Учебная степень, звание	Подпись	Дата
АиКС	Фадеев А.С.	к.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результатов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт электронного обучения

Направление подготовки (специальность) программное обеспечение вычислительной техники и автоматизированных систем (230105)

Кафедра автоматики и компьютерных систем

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Дипломного проекта

Студенту:

Группа	ФИО
3-8001	Савельеву Николаю Михайловичу

Тема работы:

Информационная система учета элементной базы лаборатории студенческого научного творчества

Утверждена приказом директора (дата, номер)

№2917/с от 15 апреля 2016г.

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

Спроектировать и реализовать информационную систему учета для элементной базы лаборатории студенческого научного творчества; система должна обеспечивать функционал необходимый для обработки хранимых в ней данных (студент, проект, деталь и различные функциональные связи между ними);

Перечень подлежащих исследованию, проектированию и разработке вопросов	1) Исследование возможностей технологии ASP.NET MVC для разработки распределенных информационных систем; 2) Исследование предметной области для построения ее информационной модели; 3) Проектирование структуры базы данных, описывающей предметную область; 4) Разработка пользовательских интерфейсов для взаимодействия с информационной системой;
Перечень графического материала	Презентация в формате *.pptx на 15 слайдах.

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Социальная ответственность	Невский Егор Сергеевич

Названия разделов, которые должны быть написаны на русском и иностранном языках:

– Заключение

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Учтая степень, звание	Подпись	Дата
Ассистент	Мартынов Я.А.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8001	Савельев Николай Михайлович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
3-8001	Савельеву Николаю Михайловичу

Институт	ИнЭО	Кафедра	АиКС
Уровень образования	Специалитет	Направление/специальность	230105

5. Социальная ответственность.	5. Предназначение раздела социальной ответственности.
5.1. Анализ возможных сбоев и их последствий.	<p>5.1. Описание различных вероятных сбоев и их возможных последствий.</p> <p>Сбои при потере соединения между компьютерами и их последствия.</p> <p>Сбои при нештатном отключении сервера и последствия.</p> <p>Сбои из-за действий пользователей.</p> <p>Возможные последствия</p> <p>Некорректная работа системы при несоответствии внесенных в систему данных реальным. Вероятные последствия.</p>
5.2. Встроенная система защиты от сбоев.	<p>5.2. Для того чтобы нивелировать возможные последствия различных сбоев в работе системы, необходимо предусмотреть максимальное количество факторов, влияющих на возникновение этих сбоев, и постараться защитить систему от их влияния.</p> <p>Защита системы от некоторых очевидных моментов, могущих привести к сбоям в работе всей системы.</p> <p>Валидация данных для отсека ввода заведомо некорректных данных. Проверка вводимых данных на соответствие требованиям к этим данным.</p> <p>Валидация предотвращает ввод только очевидно неподходящих данных и не обеспечивает полной защиты от ввода ошибочной информации.</p> <p>Различные проверки при совершении действий с данными. Дополнительное подтверждение – защита от случайного нажатия.</p>
5.3. Мероприятия по защите оборудования.	<p>5.3. Информационная система – чисто программный комплекс, значит воздействие факторов, влияющих на физическую платформу, невозможно предотвратить средствами самой системы. Для защиты необходимо обеспечить платформу физическими средствами, которые позволят игнорировать данные факторы.</p> <p>Установка источника бесперебойного питания (ИБП) как защита от возможных сбоев в питании и скачков напряжения в сети.</p> <p>Обезопасить соединение между компьютерами для доступности системы возможно при помощи защиты сетевого кабеля. Повреждение кабеля возможно предотвратить путем его прокладки в специальный</p>

	<p>кабель-канал. Не гарантирует его 100%-ой защищенности, но снижает вероятность повреждения.</p> <p>Сетевое оборудование, обеспечивающее соединение между компьютерами, так же должно быть подключено к ИБП для исключения возможности потери данных.</p>
5.4. Мероприятия по защите достоверности данных.	<p>5.4. Поскольку полностью предотвратить ввод ошибочных данных программными средствами невозможно, так как здесь присутствует человеческий фактор, то для управления системой и контроля достоверности данных необходимо назначение администратора.</p> <p>Администратор и обычные пользователи имеют различные права и возможности в системе, что подразумевает собой и различные степени ответственности.</p> <p>Проверка наличия достаточных прав для совершения действия.</p> <p>Администратор должен четко осознавать свои обязанности и надлежащим образом выполнять их.</p> <p>Исполнительность и ответственность администратора.</p> <p>На администратора возлагается обязанность по отслеживанию количества свободных деталей различных типов.</p> <p>Невнимательность администратора системы так же может привести и к безвозвратной утере каких-либо элементов.</p> <p>Пользователей системы необходимо ознакомить с правилами пользования информационной системой для снижения рисков возникновения несоответствия данных.</p> <p>Анонимность, регистрация и вход в систему.</p> <p>Ответственность администратора.</p>

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Учебная степень, звание	Подпись	Дата
Ассистент	Невский Е.С.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
З-8001	Савельев Николай Михайлович		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОС-
БЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
3-8001	Савельеву Николаю Михайловичу

Институт	Электронного обучения	Кафедра	АиКС
Уровень образования	Специалитет	Направление/специальность	230105

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала инженерных решений (ИР)	
2. Формирование плана и графика разработки и внедрения ИР	
3. Обоснование необходимых инвестиций для разработки и внедрения ИР	
4. Составление бюджета инженерного проекта (ИП)	
5. Оценка ресурсной, финансовой, социальной, бюджетной эффективности ИР и потенциальных рисков	

Перечень графического материала (с точным указанием обязательных чертежей)

1. «Портрет» потребителя
2. Оценка конкурентоспособности ИР
3. Матрица SWOT
4. Модель Кано
5. ФСА диаграмма
6. Оценка перспективности нового продукта
7. График разработки и внедрения ИР
8. Инвестиционный план. Бюджет ИП
9. Основные показатели эффективности ИП
10. Риски ИП

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8001	Савельев Николай Михайлович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт электронного обучения

Специальность программное обеспечение вычислительной техники и автоматизированных систем (230105)

Уровень образования – специалист

Кафедра автоматики и компьютерных систем

Период выполнения – весенний семестр 2016 учебного года

Форма представления работы:

Дипломный проект

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
29.02.2016г.	Анализ предметной области	15
20.03.2016г.	Проектирование информационной системы	15
17.05.2016г.	Техническая реализация	30
02.06.2016г.	Оформление результатов	10
21.05.2016 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
24.05.2016 г.	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Учебная степень, звание	Подпись	Дата
Ассистент	Мартынов Я.А.			

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Учебная степень, звание	Подпись	Дата
АиКС	Фадеев А.С.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа 76 с., 20 рис., 18 табл., 10 источников, 0 прил.

Ключевые слова: информационная система, лаборатория студенческого научного творчества, учет, ASP .NET MVC, кафедра АиКС, база данных, web-приложение.

Объектом исследования являются деятельность лаборатории студенческого творчества, предметная область элементной базы лаборатории, внутренние процессы лаборатории.

Цель работы – оптимизация ведения учета элементной базы лаборатории студенческого научного творчества путем разработки информационной системы учета.

В процессе исследования проводились изучение, систематизация и категоризация номенклатуры элементной базы лаборатории; изучение процесса ведения учета ресурсов, участников и разработок лаборатории.

В результате исследования построена информационная модель, описывающая сущности и виды их взаимодействия между собой; разработана информационная система учета.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: клиент-серверное приложение с web-интерфейсом разработанное с использованием шаблона MVC.

Степень внедрения: готова к внедрению.

Область применения: учет материальной базы лаборатории.

Экономическая эффективность/значимость работы выражается в оптимизации временных затрат на ведение учета, планирование и реализацию студенческих проектов; повышении качества образовательных услуг лаборатории.

В будущем планируется автоматизация процесса ведения учета.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	12
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	14
1.1 Постановка проблемы	15
1.2 Анализ существующих решений	17
2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	21
2.1 Классы и характеристики пользователей	21
2.2 Функциональность системы	22
2.3 База данных	23
2.4 Обоснование технологий и средств разработки	25
3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ	27
3.1 Модели	28
3.2 Контроллеры	29
3.3 Представления	32
3.4 Entity Framework	34
3.5 Внешний вид приложения	37
3.6 Диаграммы системы	37
4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ	40
4.1 Просмотр списков	40
4.2 Возможности администратора	44
5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА	47
5.1 Обоснование необходимости и актуальности разработки	47
5.2 Организация и планирование комплекса работ	48
5.2.1 Продолжительность этапов работ и календарный план	49
5.2.2 Календарный план-график работ	52
5.2.3 Расчет накопления готовности проекта	55
5.3 Расчет сметы затрат на разработку	56
5.3.1 Расчет затрат на материалы	56
5.3.2 Расчет по статье заработная плата	57
5.3.3 Отчисления на социальные нужды	59
5.3.4 Расчет затрат на электроэнергию	59
5.3.5 Расходы на оборудование для выполнения работ	60
5.3.6 Расчет амортизационных расходов	60
5.3.7 Расходы на услуги сторонних организаций	62
5.3.8 Прочие расходы	62

5.3.9 Расчет общей себестоимости.....	62
5.3.10 Расчет прибыли.....	62
5.3.11 Расчет НДС.....	63
5.3.12 Цена разработки НИР.....	63
5.4 Оценка экономической эффективности проекта	63
5.5 Оценка научного уровня.....	64
6 Социальная ответственность.....	67
6.1 Анализ возможных сбоев и их последствий.....	67
6.2 Встроенная защита системы от сбоев	69
6.3 Мероприятия по защите оборудования.....	71
6.4 Мероприятия по защите достоверности данных.....	72
ЗАКЛЮЧЕНИЕ.....	76
CONCLUSION.....	77
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	78

ВВЕДЕНИЕ

В связи с постоянным развитием информационных технологий различные образовательные учреждения зачастую не успевают подготавливать необходимых специалистов, которые могли бы применять эти технологии на практике. Это связано в первую очередь с тем, что учебная программа довольно статична, и нет возможности изменить ее в течение учебного года. Поэтому все чаще стали появляться различные лаборатории, не вписывающиеся в обязательную программу обучения, но дающие возможность освоить различные новые методы и технологии, причем не только в теории, но и на практике.

Институт кибернетики Томского политехнического университета так же не остался в стороне от данной тенденции, в результате чего на базе кафедры автоматики и компьютерных систем была создана лаборатория студенческого научного творчества.

Целью данной выпускной квалификационной работы является исследование деятельности лаборатории студенческого научного творчества и протекающих в ее рамках процессов для построения эффективной информационной системы учета элементной базы лаборатории. Для этого необходимо детально изучить номенклатуру элементной базы, систематизировать ее и предусмотреть различные сценарии по взаимодействию с частями этой базы. Кроме того, в процессе наблюдения за работой лаборатории необходимо изучить различные функциональные связи между всеми элементами предметной области лаборатории.

На основе данных, полученных при исследовании лаборатории, построить информационную модель данных, отражающую основные параметры элементов предметной области и описывающую всевозможные их взаимосвязи.

Используя полученную модель, разработать информационную систему для лаборатории, максимально точно отражающую параметры элементов предметной области и предоставляющую необходимый функционал для совершения всех требуемых действий над ними.

Результатом должна стать система, готовая к внедрению в структуру лаборатории для оптимизации протекающих в ней процессов и структурирования хранящихся в ней данных.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В последнее время все заметнее становится всевозрастающий интерес развивающихся предприятий реального сектора экономики к специалистам, владеющим технологиями проектирования, интеграции, настройки современных встраиваемых сверхкомпактных промышленных систем, а также разработки для них прикладного программного обеспечения и доведения готового изделия до опытного тиражируемого образца. Основным достоинством таких систем является высокая интеграция, достигаемая за счет изготовления всего компьютера на одной плате размером, как правило, с кредитную карту, которая приводит к чрезвычайно высокой ресурсоэффективности как самого встроенного компьютера, так и всей системы в целом. При этом функционал, программные и аппаратные возможности таких систем высокими темпами приближаются к возможностям не столь ограниченных ресурсами персональных компьютеров. Все чаще встраиваемые сверхкомпактные системы начинают активно интегрироваться в мобильные и подвижные устройства, бытовые приборы, медицинское и промышленное оборудование, военное оборудование и оборудование двойного назначения. Все это, а так же, помимо всего прочего, их ценовая доступность, подогревает интерес к данным системам в среде обучающихся.

В связи с данными тенденциями на базе кафедры автоматики и компьютерных систем института кибернетики была создана лаборатория студенческого научного творчества. Проект данной лаборатории нацелен на развитие практических навыков обучающихся в сфере конструирования промышленного оборудования на основе передовых сверхкомпактных компьютерных технологий мирового уровня для повышения интеллектуальной составляющей технических систем и их конкурентоспособности.

В основе проекта лаборатории лежат три компоненты:

- высокотехнологичная, доступная обучающимся, современная инфраструктура, базирующаяся на сверхкомпактных встраиваемых компьютерах;
- банк задач реального сектора экономики, основанных на потребности промышленности в повышении интеллектуальной составляющей производимого оборудования сверхкомпактными средствами контроля и управления;
- профессорско-преподавательский состав, работающий над проектами и по темам НИР предприятий реального сектора экономики.

В рамках проекта по организации лаборатории было произведено оснащение рабочих мест обучающихся комплектами современных встраиваемых сверхкомпактных компьютеров и вспомогательным оборудованием, позволяющим осуществлять интеграцию сверхкомпактных компьютеров в оборудование широкого потребления, его комплексную настройку, отладку, разработку прикладного программного обеспечения и доведение готового изделия до опытного образца. Средства на создание данной лаборатории были получены благодаря гранту благотворительного фонда Потанина.

1.1 Постановка проблемы

Самообразование в учебных заведениях всегда было неотъемлемой частью учебного процесса. Теоретические и практические курсы жестко регламентируются программой обучения. Потому, некоторые обширные темы невозможно охватить целиком в учебное время во время пар или уроков. В этих случаях подразумевается, что необходимый материал студенты или ученики будут осваивать самостоятельно, используя рекомендованную литературу и иные доступные источники информации. Кроме того, всегда были, есть и будут темы, интересные конкретному человеку, которые он стремится изучить для саморазвития, в связи с выбранной будущей профессией, либо просто в качестве хобби. Однако не все теоретические знания учащиеся мо-

гут подкрепить необходимой практикой у себя дома из-за отсутствия необходимой аппаратуры. Например, программисты, интересующиеся программированием микроконтроллеров, не всегда могут на практике проверить работоспособность той или иной своей программы. В связи с таким положением дел и, идя навстречу обучающимся, в институтах и на кафедрах открываются лаборатории, предоставляющие необходимое оборудование для занятия исследовательской деятельностью.

Такой стала и лаборатории студенческого научного творчества, открывшаяся на базе кафедры АиКС Института кибернетики Томского политехнического университета. Лабораторией закупаются все необходимые компоненты (как электронные и электрические, так и механические) для успешного создания управляемых автоматизированных систем на базе микроконтроллеров. С момента ее открытия число студентов, желающих попробовать воплотить свои идеи в жизнь, только увеличивается, что позволяет начинающим программистам получить бесценный в нынешнее время практический опыт.

С ростом количества научных проектов и вовлеченных в них студентов, расширяется и сама лаборатория, закупая все больший ассортимент всевозможных деталей, которые могут понадобиться в процессе деятельности участников лаборатории.

В настоящий момент в лаборатории насчитывается порядка 30 категорий различных компонентов, около сотни наименований и гораздо большее их количество. Данные по всем компонентам (их характеристики, места их хранения и т.д.) систематизированы в несколько журналов, процесс работы с которыми весьма не удобен и долговат. Учет использующихся и не задействованных деталей происходит путем занесения в отдельный рукописный журнал той или иной записи, что так же занимает достаточно большое количество времени. Плюсом к этому идут различные проблемы с возможностью уте-

ри этих данных, либо невозможностью их чтения (плохой почерк, сокращения, недостаточность времени для добавления необходимых записей). Кроме того, возможностью добавлять записи в журнал обладают только преподаватели, имеющие отношение к лаборатории, у которых не всегда есть время находиться в ней в свое рабочее время (наличие пар в расписании, выполнение своих служебных обязанностей и т.д.).

Все эти факторы, в свою очередь, негативно влияет на скорость развития студенческих проектов. Налицо узкое место в системе организации работы лаборатории.

1.2 Анализ существующих решений

В результате исследования лаборатории были получены результаты, на основании которых можно сделать вывод – лаборатории необходима система, которая бы позволила облегчить и ускорить процесс учета элементной базы и выполняемых проектов, и сделала бы ее более доступной.

Похожие задачи довольно часто встречаются в различных предприятиях, которые в той или иной степени используют в процессе своей жизнедеятельности некую достаточно большую материальную базу. Им необходим удобный инструмент для учета различных элементов, составляющих саму базу.

Для решения таких задач используются информационные системы, позволяющие хранить, обрабатывать и выдавать информацию, отображающую некую реально существующую систему. Информационная система позволяет надлежащим людям своевременно получать необходимую информацию в рамках определенной предметной области. Такие системы в своей основе имеют три важных принципа: принцип интеграции, принцип системности и принцип комплексности.

Первый принцип заключается в том, что данные, внесенные в систему, могут многократно использоваться в процессе решения различных задач. Второй – подразумевает, что данные могут рассматриваться в различных аспектах. И третий – что данные автоматически преобразуются на всех этапах работы системы.

В данный момент на рынке находится огромное количество всевозможных информационных систем, которые оптимизируют те или иные процессы. Например, информационная система по отысканию рыночных ниш или информационная система по снижению издержек производства. В случае лаборатории студенческого научного творчества наиболее близкими аналогами будут являться всевозможные информационные системы учета. Однако при близком рассмотрении становится понятно, что все они либо весьма узкоспециализированы, что не позволит использовать их для других предметных областей, отличных от той, для которой они были разработаны, либо предназначены для использования в коммерческой деятельности и включают в себя возможность расчета различных экономических показателей.

Например, система онлайн учета для бизнеса Большая Птица. Эта система позволяет вести складской учет товаров в торговле. Вот ее основные возможности, как **складской программы** (внешний вид программы представлен на рисунке 1):

- а) Учет товаров по нескольким складам;
- б) Поступление и расход товаров со склада в результате торговых операций;
- в) Проведение инвентаризации склада целиком, либо отдельных товарных групп и позиций;
- г) Оприходование товаров в случае обнаружения на складе их излишков, не учтенных в программе;

д) Списание товаров при установлении факта недостачи остатков номенклатуры на складе;

е) Перемещение товаров между разными складами компании.

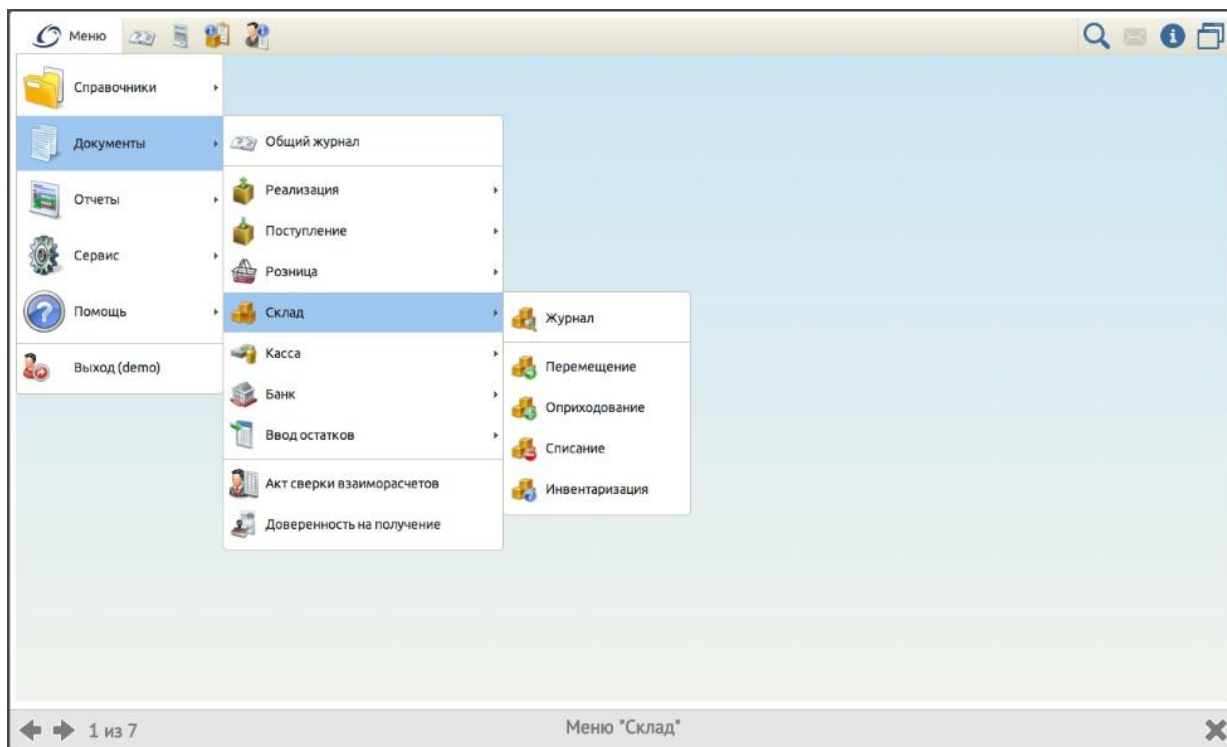


Рисунок 1 – Внешний вид программы Большая Птица

Или интернет сервис под названием МойСклад, аналог Большой Птицы. Она включает в себя возможность складского учета, розничной торговли, оптовой продажи и электронной коммерции, составления аналитических отчетов и т.д. Внешний вид этой программы представлен на рисунке 2.

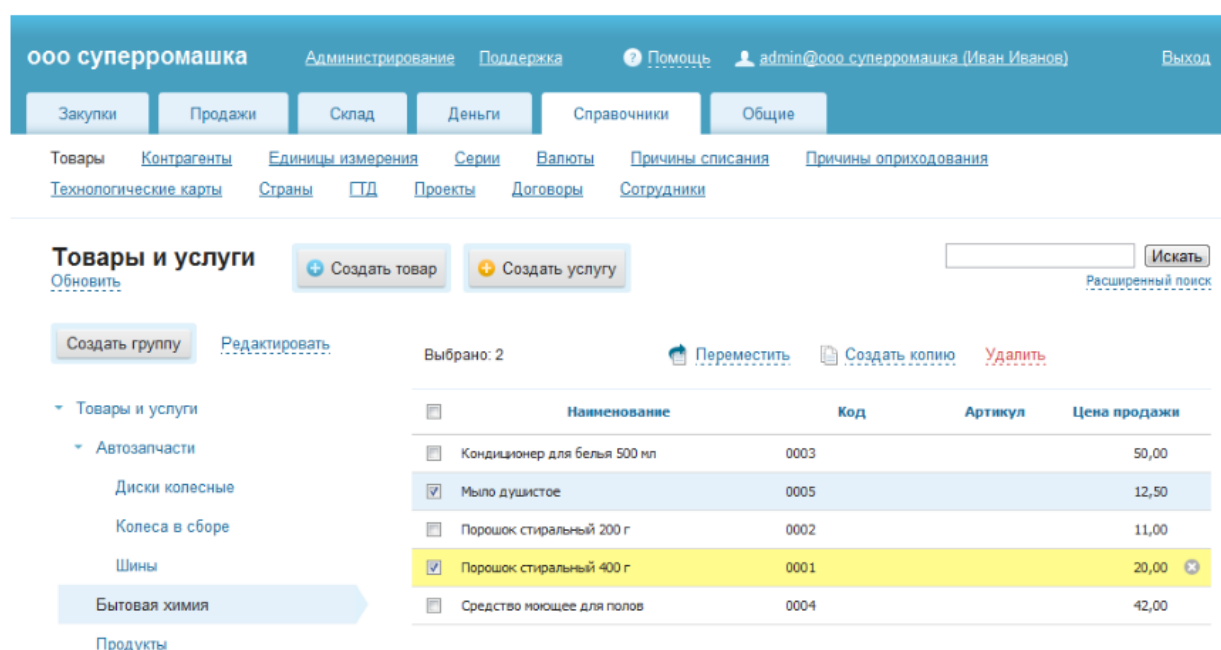


Рисунок 2 – Внешний вид программы МойСклад

Как видно из описания возможностей, данные программы рассчитаны на использование компаниями, занимающимися торговой деятельностью. Применительно же к потребностям лаборатории подавляющее большинство функций этих программ останется не востребуемым. Кроме того, данные продукты являются коммерческими, следовательно, у них нет бесплатных версий, предназначенных для различных нужд некоммерческих организаций.

Таким образом, поскольку на рынке нет программных продуктов, подходящих под нужды лаборатории, целесообразным становится разработка такого продукта собственными силами.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Информационная система учета элементной базы лаборатории студенческого научного творчества предназначена для работы со студенческими проектами. Соответственно структура хранимых данных должна быть выстроена таким образом, чтобы существовала возможность просмотреть информацию о студентах, их проектах и использующихся в них деталях, отражающую их текущее состояние. Кроме того, система должна обладает возможностью регистрации/авторизации пользователей с дальнейшим изменением своего состояния.

2.1 Классы и характеристики пользователей

В таблице 1 приведен список пользователей системы.

Таблица 1 – Виды пользователей системы

Наименование	Краткое описание
Анонимный пользователь	Пользователь, не прошедший регистрацию и авторизацию в системе. Имеет права, достаточные только для ознакомления с хранящейся в системе общедоступной информацией. Не имеет возможности изменять, добавлять или удалять данные.
Зарегистрировавшийся пользователь	Пользователь, прошедший процесс регистрации и авторизации в системе. Имеет те же права, что и анонимный пользователь, плюс возможность создавать и изменять некоторые записи в системе.
Администратор системы	Пользователь, вошедший в систему под учетной записью администратора. Имеет полный доступ ко всей информации.

2.2 Функциональность системы

Так как система работает с несколькими основными и несколькими вспомогательными сущностями, то присутствует и необходимый набор функций для работы с данными. Основными функциями всегда являются добавление, просмотр, изменение, удаление. В процессе реализации системы возникла потребность в дополнительном функционале.

Для более полного описания необходимого функционала системы составлена диаграмма вариантов использования, представленная на рисунке 3.

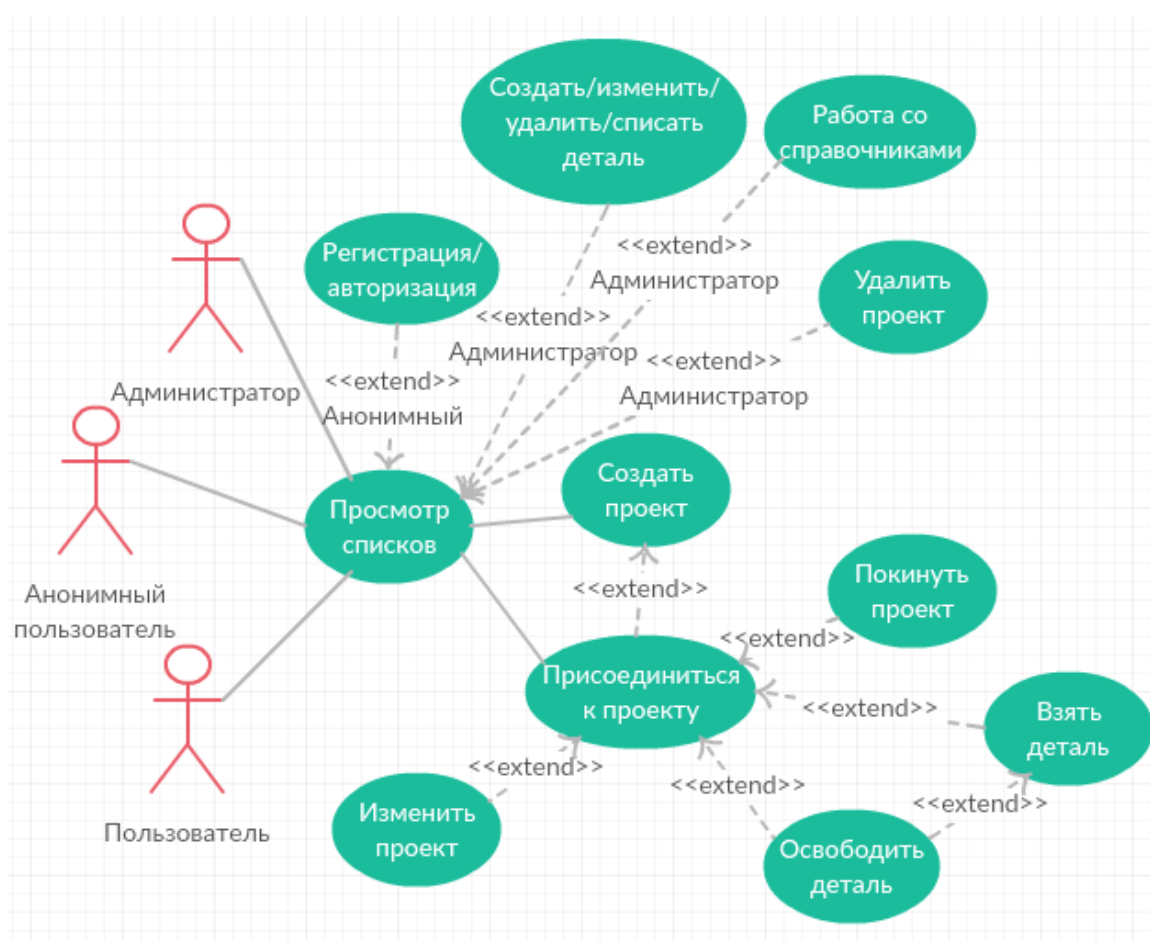


Рисунок 3 – Диаграмма вариантов использования (use case diagram)

Диаграмма вариантов использования описывает функциональность системы и доступность ее функций для различных типов пользователей в зависимости от уже выполненных действий.

2.3 База данных

Система должна быть рассчитана на работу с несколькими основными и несколькими вспомогательными сущностями. Основные сущности это – студент, проект и деталь. Вспомогательные – категория детали, списанная деталь, учебная группа, статус проекта и промежуточная сущность связи многие-ко-многим между сущностями проект и студент.

Концептуальная модель, описывающая взаимосвязи основных сущностей, изображена на рисунке 4.



Рисунок 4 – Взаимосвязи сущностей проект, деталь и пользователь

Экземпляр сущности типа деталь при занесении в базу данных должен будет включать в себя информацию о ее наименовании, количестве и некоторую другую вспомогательную информацию. Кроме того, каждый экземпляр должен обладать возможностью быть привязанным к нескольким проектам, что указывает на необходимость хранения списка связанных проектов для каждого экземпляра.

Основной информацией для проекта является название и дата начала проекта.

Поскольку проект может использовать в себе сразу несколько различных деталей, так же как и одна деталь одного типа может быть использована в нескольких проектах одновременно, возникает необходимость связи многие-ко-многим, реализация которой возможна путем создания промежуточной сущности для хранения списка связей. Экземпляры этой сущности так же должны хранить количество деталей определенного типа, задействованных в конкретном проекте.

Так как в системе предусмотрена возможность вступления пользователя сразу в несколько проектов, а одним проектом могут заниматься несколько человек, то для связи экземпляров этих двух сущностей так же необходима промежуточная сущность.

Сущность типа пользователь должна предусматривать возможность хранения личной информации типа фамилии, имени и отчества.

Для более четкой систематизации данных и их упрощенной фильтрации в дальнейшем в системе необходимо предусмотреть наличие справочников:

- а) справочник категорий деталей;
- б) справочник статусов проектов;
- в) справочник учебных групп.

Справочник категорий необходим, поскольку различных деталей в лаборатории хранится достаточно большое количество, и для поиска необходимой детали, либо ее аналога, его использование выгодно.

То же самое относится к проектам. Поскольку в системе могут храниться уже завершенные, либо временно замороженные проекты, существует необходимость их как-то отличать от тех, что находятся в активной разработке. Для этого необходим ввод справочника статусов.

Справочник учебных групп необходим для более точной персонализации пользователя. Это может пригодиться при необходимости разыскать какого-то конкретного студента, что может быть сделано через расписание занятий.

Подробная структура базы данных приведена на рисунке 5.

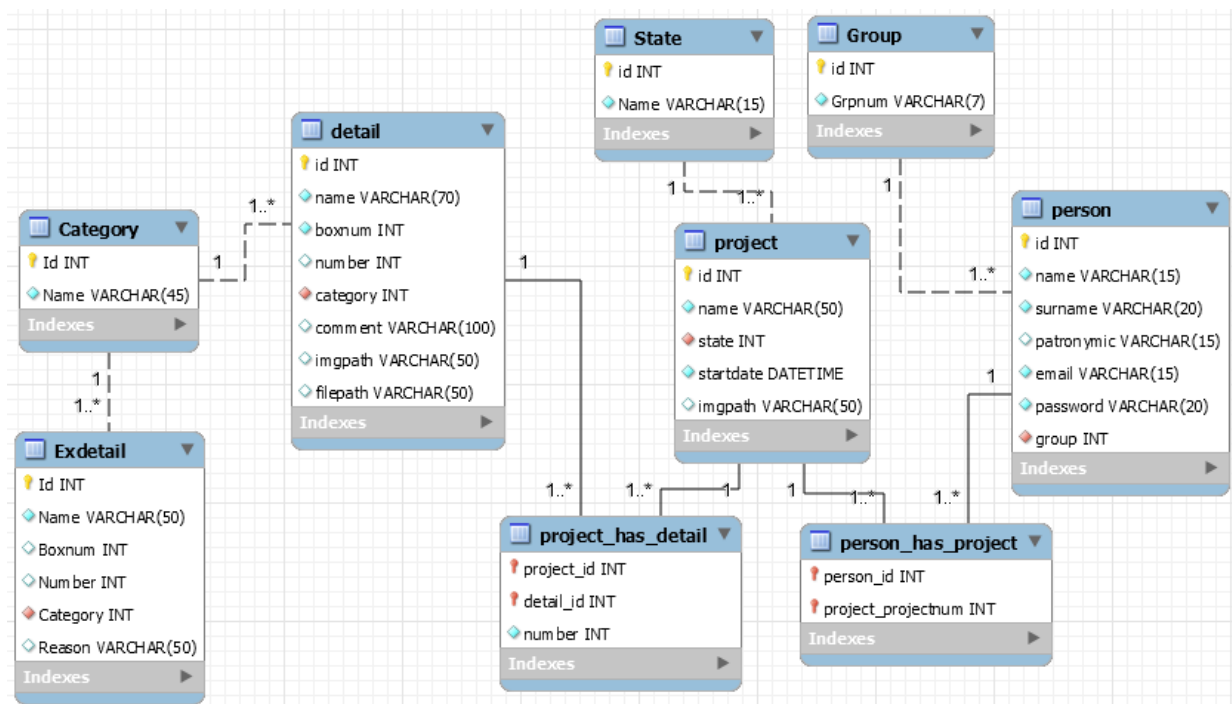


Рисунок 5 – Схема базы данных

Как видно из рисунка, у каждой сущности помимо основных атрибутов есть дополнительные, так, для детали это номер хранилища, адрес для прикрепленного изображения и адрес для прикрепленного файла спецификации. Для сущности пользователь – пароль для доступа к учетной записи и email, являющий одновременно и логином, и дающий возможность связи со студентом.

2.4 Обоснование технологий и средств разработки

В лаборатории расположено несколько рабочих мест, оборудованных ПК, с которых должна быть возможность работы с системой. В данном случае, идеально подойдет клиент-серверная информационная система. То есть, данные будут храниться централизованно (на сервере), но доступ к ним можно будет получить с любого находящегося в сети компьютера и в любое время. Поскольку количество ПК, которые будут иметь доступ к серверу, не слишком велико, то даже обычный компьютер вполне сможет послужить платформой для установки системы и обеспечить все ее потребности в плане

обработки входящих запросов. Это, в свою очередь, отменяет необходимость покупки мощного дорогостоящего сервера для размещения системы.

В большинстве подходов клиент-серверное приложение представляет собой систему, состоящую из четырех компонент – база данных, СУБД, серверное приложение и клиентское приложение. Или, в случае web-приложения – база данных, СУБД и web-сервер. Зачастую, каждая из этих компонент – отдельная разработка, и чтобы заставить их работать вместе приходится прикладывать значительные усилия.

В качестве информационной системы для лаборатории лучше всего подойдет web-приложение. Оно позволит работать с данными из любой аудитории учебного корпуса, не требуя при этом установки специального клиентского программного обеспечения (ПО).

До недавнего времени процесс разработки web-приложения был условно разбит на три разные части: разработка базы данных и написание запросов к ней, разработка интерфейса путем составления html страниц и написание скриптов для взаимодействия интерфейса и базы данных. Однако несколько лет назад, в 2009 году, появилась технология, способная объединить процессы разработки каждой из частей web-приложения в один общий проект для лучшего их согласования – ASP .NET MVC.

Эта платформа предназначена для создания сайтов и web-приложений с использованием паттерна (или шаблона) MVC (model - view - controller). Именно эта платформа 5-ой версии была выбрана для разработки информационной системы учета элементной базы лаборатории студенческого научного творчества.

Использование технологии MVC автоматически определяет выбор Microsoft Visual Studio в качестве среды разработки (в данном случае – Microsoft Visual Studio Community 2013), а MSSQL – в качестве системы управления базой данных, поскольку именно на использование этих компонент MVC и рассчитан.

3 ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

Концепция шаблона MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) – это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

Общая схема взаимодействия этих компонентов изображена на рисунке 6.

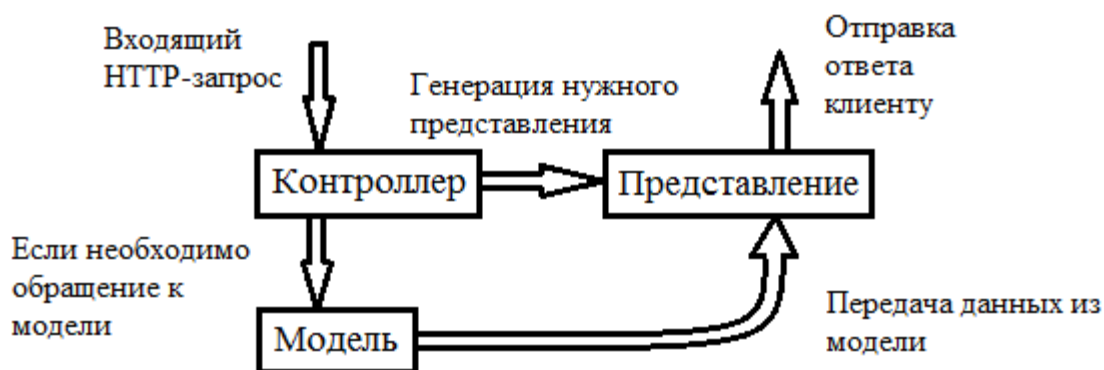


Рисунок 6 – Схема взаимодействия шаблона MVC

В этой схеме модель является независимым компонентом - любые изменения контроллера или представления не затрагивают модель. Контроллер и представление являются относительно независимыми компонентами, и нередко их можно изменять независимо друг от друга.

Благодаря этому реализуется концепция **разделения ответственности**, в связи с чем легче построить работу над отдельными компонентами. Кроме того, вследствие этого приложение обладает лучшей тестируемостью, поскольку контроллер и представления могут тестироваться отдельно друг от друга.

Механизм взаимодействия компонентов MVC, некоторые его особенности и простейшие примеры реализации хорошо описаны в руководстве по ASP .NET MVC 5[1]. Однако для получения глубокого понимания не обойтись без изучения полноценных литературных источников[2]. Кроме того, необходимо иметь хорошее представление об особенностях языка C#[3] и технологии .NET[4].

3.1 Модели

Модели представляют собой простые классы и располагаются в проекте в каталоге *Models*. Модели описывают логику данных. Например, модель, представляющая основные параметры детали:

```
public class Detail
{
    //Id записи
    public int Id { get; set; }
    //Наименование детали
    public string Name { get; set; }
    //Количество деталей
    public int? Number { get; set; }
    //Набор возможных связей с проектами
    public ICollection<Detailprojconnection> Detailprojconnections { get;
set; }
    public Detail()
    {
        Detailprojconnections = new List<Detailprojconnection>();
    }
}
```

Каждая модель описывает какую-то сущность базы данных, какие именно данные будут храниться для сущности, и каким образом экземпляры различных сущностей будут взаимодействовать.

Поскольку во время разработки системы использовался подход *code first*, то при первой компиляции проекта на основе моделей генерируются *sql*-запросы, формирующие структуру базы данных. До этого момента база остается пустой, поэтому модели должны быть четко проработаны заранее, чтобы сразу сформировать верную структуру базы.

Модель необязательно состоит только из свойств, кроме того, она может иметь конструктор, какие-нибудь вспомогательные методы. Однако предназначение модели – именно описывать данные. Манипуляции с данными и бизнес-логика – это больше сфера контроллера.

Всего в системе 8 моделей, на одну меньше, чем в спроектированной базе данных. Так как MVC автоматически генерирует промежуточные сущности для связей типа многие-ко-многим, то необходимость создавать данные модели вручную отпадает, за исключением тех случаев, когда этой сущности необходимо придать дополнительные атрибуты. В данном случае, для промежуточной сущности связи между проектом и деталью необходимо было вручную прописать модель из-за дополнительного поля, отображающего количество задействованных в проекте деталей.

3.2 Контроллеры

Контроллер является центральным компонентом в архитектуре MVC. Контроллер получает ввод пользователя, обрабатывает его и посылает обратно результат обработки, например, в виде представления.

Контроллер представляет обычный класс, который наследуется от базового класса **System.Web.Mvc.Controller**. В свою очередь класс **Controller** реализует абстрактный базовый класс **ControllerBase**, а через него и интерфейс **IController**. Таким образом, формально, чтобы создать свой класс контроллера, достаточно создать класс, реализующий интерфейс **IController** и имеющий в имени суффикс *Controller*.

Интерфейс `IController` определяет один единственный метод `Execute`, который отвечает за обработку контекста запроса:

```
public interface IController
{
    void Execute(RequestContext requestContext);
}
```

При обращении к любому контроллеру система передает в него контекст запроса. В этот контекст запроса включается все: куки, отправленные данные форм, строки запроса, идентификационные данные пользователя и т.д.

Хотя с помощью реализации интерфейса `IController` очень просто создавать контроллеры, но в реальности чаще оперируют более высокоуровневыми классами, как например класс `Controller`, поскольку он предоставляет более мощные средства для обработки запросов. И если при реализации интерфейса `IController` мы имеем дело с одним методом `Execute`, и все запросы к этому контроллеру, будут обрабатываться только одним методом, то при наследовании класса `Controller` мы можем создавать множество методов действий, которые будут отвечать за обработку входящих запросов, и возвращать различные результаты действий.

Весь функционал приложения определяется наборами методов контроллеров. В разработанном приложении количество контроллеров равняется количеству моделей, то есть 8-ми.

Контроллер проектов (`ProjectsController`) обеспечивает вывод списка всех проектов, просмотр детальной информации о любом проекте, создание/редактирование/удаление проекта, возможность присоединиться к проекту или покинуть его. Ограничение доступа к некоторым из этих функций так же обеспечивается контроллером.

Методы `Join` и `Leave` контроллера проектов обеспечивают возможность студентов присоединяться к понравившимся проектам, либо покидать

их. На вход эти методы принимают id проекта и либо создают связь студента и проекта (путем создания экземпляра промежуточной сущности, т.к. связь типа многие-ко-многим), либо удаляет ее.

Контроллер деталей (HomeController) так же позволяет вывести список деталей, подробную информацию о конкретной детали, создание/изменение/удаление детали, обеспечивает возможность взять определенные детали для проекта, а так же списать испорченные детали.

Метод Take обеспечивает возможность создать связь между проектом и деталью. Таким образом, деталь резервируется под определенный проект. На вход этот метод принимает id детали, id проекта и количество необходимых деталей (amount). В зависимости от того, существует уже подходящий экземпляр связующей сущности или нет, у этого экземпляра увеличивается количество деталей, или таковой экземпляр создается.

Метод Broke обеспечивает возможность списания деталей, вышедших из строя. На вход берется id детали, количество для списания и строка, содержащая причину списания, далее создается новый экземпляр сущности списанных деталей.

Контроллер списанных деталей (ExdetailsController) позволяет просмотреть весь список испорченных деталей и удалить определенную позицию.

Контроллер использующихся в проектах деталей (DetailprojconnectionsController) позволяет просмотреть список используемых в текущих проектах деталей и вернуть ненужные детали обратно на хранение.

Метод Release (вернуть деталь) контроллера деталей, задействованных в проектах, позволяет освободить ненужные детали из проекта. На вход принимается id связующей записи.

Контроллер категорий (CategoriesController) обеспечивает необходимый функционал для работы с категориями деталей (создание/изменение/удаление).

Контроллер учебных групп (GroupsController) обеспечивает необходимый функционал для работы с учебными группами (создание/изменение/удаление).

Контроллер статусов (StatesController) обеспечивает необходимый функционал для работы со статусами проектов (создание/изменение/удаление).

Контроллер пользователей (AppusersController) позволяет выводить список всех пользователей системы (включая списки проектов, в которых они заняты) и удалять пользователей.

3.3 Представления

Хотя работа приложения MVC управляется главным образом контроллерами, но непосредственно пользователю приложение доступно в виде представления, которое и формирует внешний вид приложения. В ASP.NET MVC 5 представления – это файлы с расширением *cshtml*, которые содержат код пользовательского интерфейса в основном на языке html. Например, стандартное представление выглядит так:

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>SomeView</title>
</head>
<body>
    <div>
        <h2>@ViewBag.Message</h2>
    </div>
</body>
```


</html>

Хотя представление содержит, главным образом, код html, оно не является html-страницей. При компиляции приложения на основе требуемого представления сначала генерируется класс на языке C#, а затем этот класс компилируется.

Для полноценного отображения необходимой информации и удобства ее ввода каждой модели соответствует набор представлений.

Не смотря на то, что стандартное представление в MVC очень похоже на обычную web-страницу, содержащую код разметки html, оно также имеет вставки кода на C#, которые предваряются знаком @. Этот знак используется движком представлений Razor для перехода к коду на языке C#. При вызове метода View контроллер не производит рендеринг представления и не генерирует разметку html. Контроллер только готовит данные и выбирает, какое представление надо вернуть в качестве объекта ViewResult. Затем уже объект ViewResult обращается к движку представления для рендеринга представления в выходной результат.

Используя эту особенность, в большинстве случаев можно обойтись без скриптовых языков. Например, для выполнения различных проверок. В таком случае страница может выглядеть совершенно по-разному в зависимости от различных условий. Пример такой проверки можно увидеть ниже.

```
@if (User.IsInRole("admin"))
{
    <li class="dropdown">
        <a id="drop2" href="#" class="dropdown-toggle" data-toggle =
"dropdown">
            Администрирование
            <span class="caret"></span>
        </a>
        <ul class="dropdown-menu">
            <li>@Html.ActionLink("Категории    деталей", "Index",
"Categories")</li>
            <li>@Html.ActionLink("Статусы    проектов", "Index",
"States")</li>
            <li>@Html.ActionLink("Учебные    группы", "Index",
"Groups")</li>
        </ul>
    </li>
}
```

```

        <li class="divider"></li>
        <li>@Html.ActionLink("Списанные детали", "Index",
"Exdetails")</li>
    </ul>
</li>
}

```

Здесь происходит проверка, является ли текущий пользователь системы администратором. Если условие проверки выполняется, то на странице отобразится выпадающее меню, содержащее пункты, доступные только администратору. В любом другом случае это меню не отобразится и даже не будет присутствовать в итоговой html разметке страницы. Таким образом, страница рендерится исходя из набора некоторых начальных условий.

3.4 Entity Framework

Данные моделей, как правило, хранятся в базе данных. Для работы с базой данных очень удобно пользоваться фреймворком **Entity Framework**, который позволяет абстрагироваться от написания sql-запросов, от строения базы данных и полностью сосредоточиться на логике приложения.

Entity Framework поддерживает подход "Code first", который предполагает сохранение или извлечение информации из БД на SQL Server без создания схемы базы данных или использования дизайнера в Visual Studio. Наоборот, разработчик создает обычные классы, а Entity Framework уже сам определяет, как и где сохранять объекты этих классов.

Для подключения к базе данных через Entity Framework, нужен посредник — **контекст данных**. Контекст данных представляет собой класс, производный от класса **DbContext**. Контекст данных содержит одно или несколько свойств типа **DbSet<T>**, где T представляет тип объекта, хранящегося в базе данных.

Entity Framework представляет собой специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с дан-

ными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне идет оперирование таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который предлагает Entity Framework, уже идет работа с объектами.

Центральной концепцией Entity Framework является понятие **сущности** или entity. Сущность представляет собой набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то мы можем выделить такие свойства, как имя, фамилия, рост, возраст, вес. Свойства не обязательно представляют простые данные типа int, но и могут представлять более комплексные структуры данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют **ключами**.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-к-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ можно не только извлекать определенные строки, хранящие объекты, из базы данных, но и получать объекты, связанные различными ассоциативными связями.

Другим ключевым понятием является **Entity Data Model**. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней: концептуального, уровень хранилища и уровень сопоставления (маппинга).

На концептуальном уровне происходит определение классов сущностей, используемых в приложении.

Уровень хранилища определяет таблицы, столбцы, отношения между таблицами и типы данных, с которыми сопоставляется используемая база данных.

Уровень сопоставления (маппинга) служит посредником между предыдущими двумя, определяя сопоставление между свойствами класса сущности и столбцами таблиц.

Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.

Entity Framework предполагает три возможных способа взаимодействия с базой данных —

а) **Database first**: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;

б) **Model first**: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;

в) **Code first**: разработчик создает класс модели данных, которые будут храниться в БД, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

Базовые принципы и особенности работы данной технологии хорошо изложены в руководстве[5].

3.5 Внешний вид приложения

Внешний вид приложения представляет собой web-интерфейс, построенный при помощи технологии ASP .NET MVC. Таким образом, за интерфейс отвечают представления приложения.

Все представления приложения разбиты по папкам в зависимости от моделей, для которых написаны. Практически все модели имеют представления для вывода всего списка записей, добавления, редактирования, просмотра деталей и удаления. Кроме того, практически каждая дополнительная функция для работы с таблицами имеет свое представление (например, присоединение к проекту или выход из него).

Помимо отображения необходимой информации, представления ограничивают доступ к некоторым функциям в зависимости от того, авторизован ли пользователь и какого типа у него учетная запись. Конечно, это ограничение легко обходится через адресную строку, но повторная проверка в контроллерах позволяет пресечь подобные попытки.

3.6 Диаграммы системы

Диаграмма последовательности позволяет отобразить жизненный цикл запроса пользователя в системе. Поскольку в MVC при различных запросах взаимодействие компонент происходит практически по одному и тому же сценарию, то диаграммы последовательностей для различных запросов будут идентичны друг другу. В общем виде процесс взаимодействия представления, контроллера и модели в MVC представлен на диаграмме последовательностей на рисунке 7. Сначала происходит вывод начальной (стартовой) страницы, далее пользователь вводит на ней некие данные, которые затем проверяются и обрабатываются. В конце происходит вывод результатов.

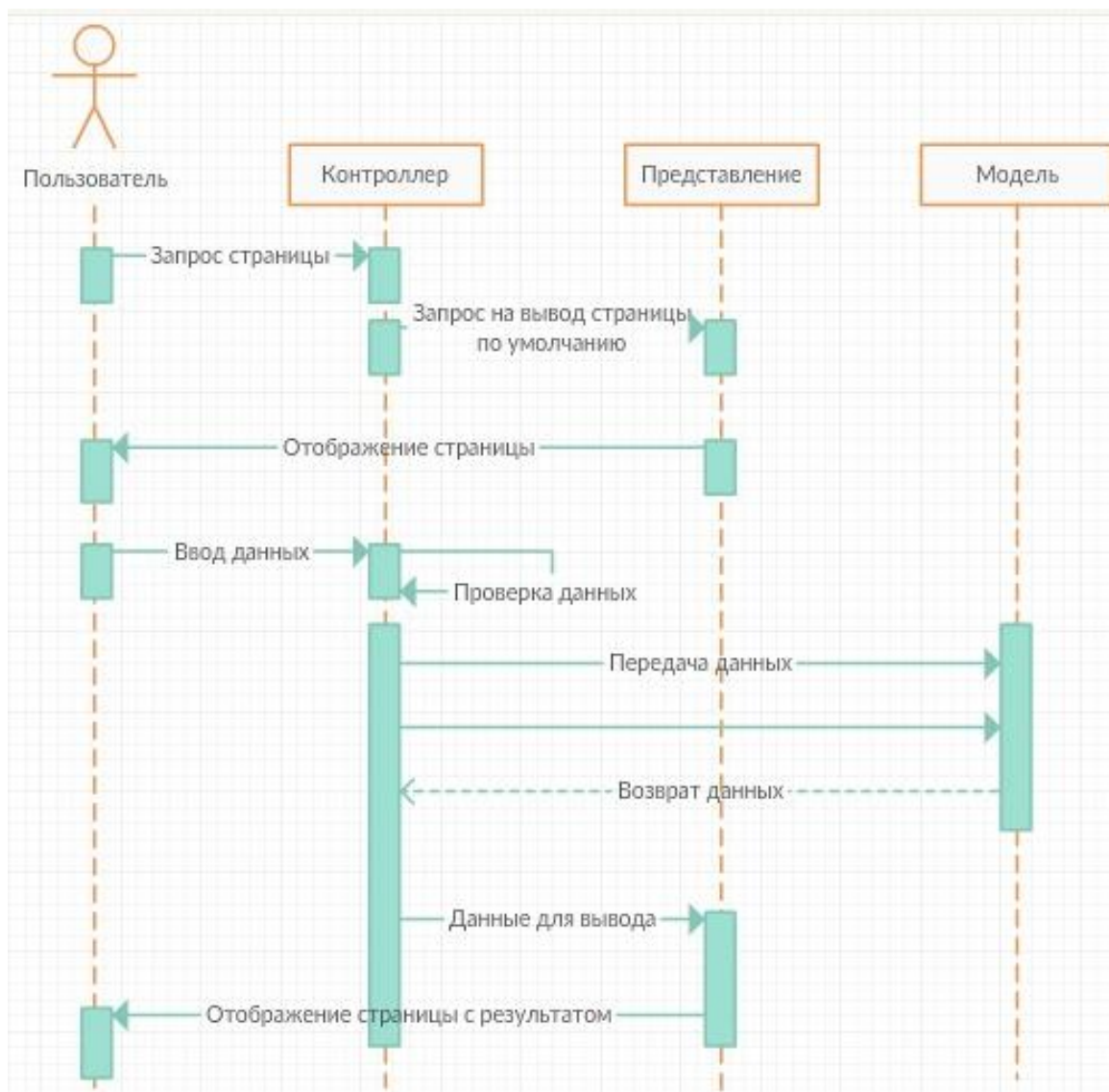


Рисунок 7 – Диаграмма последовательностей (sequence diagram)

Диаграммы классов, сгруппированные по моделям данных, с которыми они работают, приведены на рисунках 8 и 9 и демонстрируют атрибуты и методы классов.

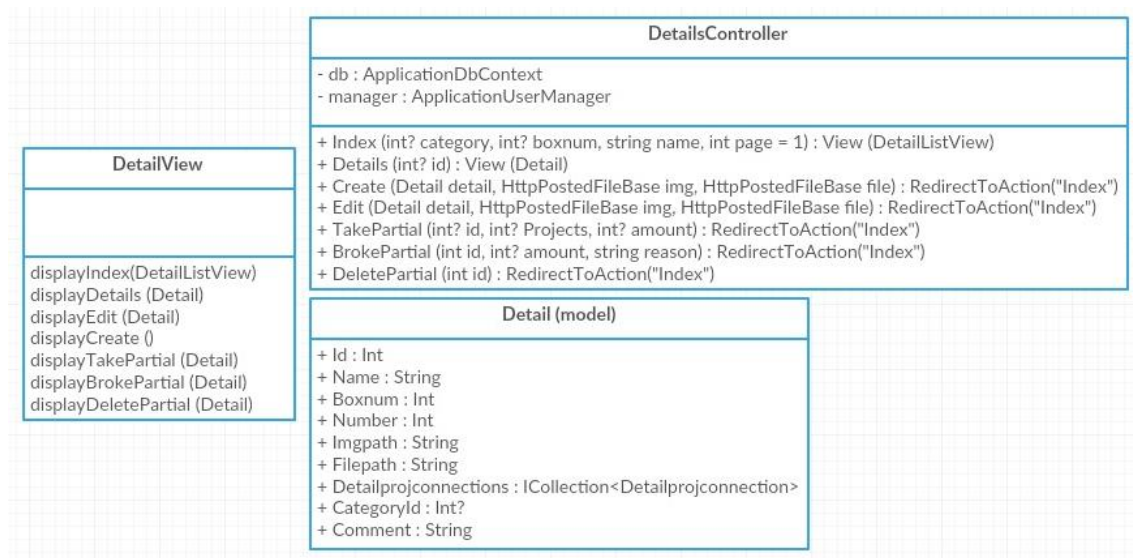


Рисунок 8 – Диаграмма классов для сущности деталь (Detail)

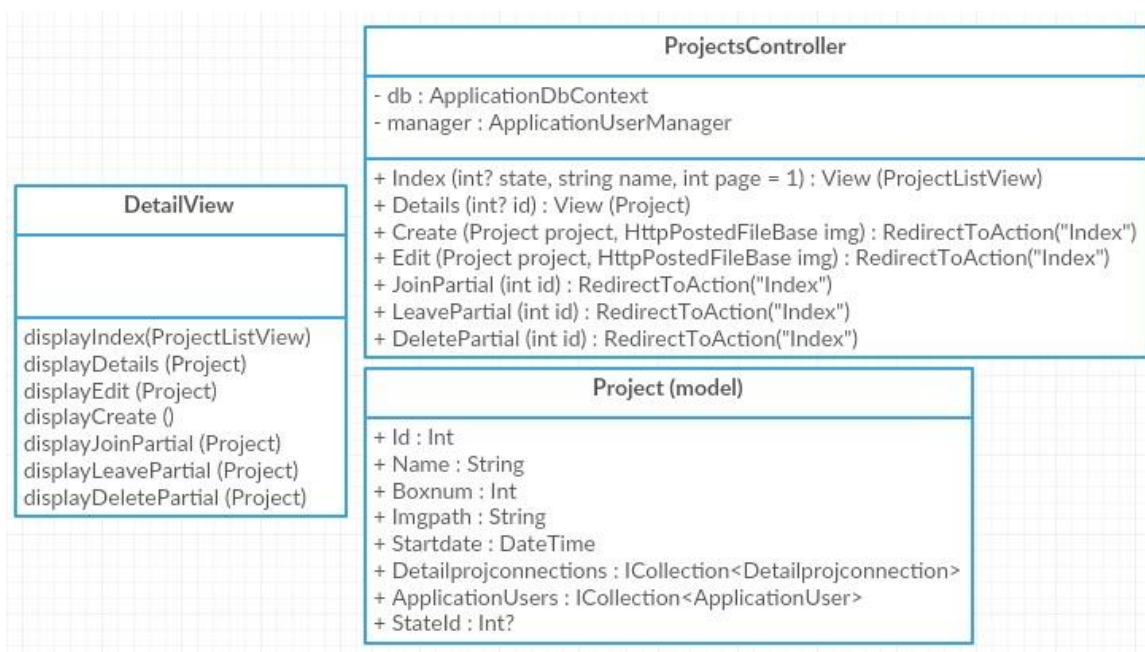


Рисунок 9 – Диаграмма классов для сущности проект (Project)

Поскольку в большинстве случаев, контроллеры и представления специализированы для работы с конкретной моделью, то такая группировка вполне обоснована. Однако внутри одного контроллера могут вызываться методы другого контроллера, так же как и представления одного контроллера могут содержать ссылки на методы стороннего контроллера.

4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ

4.1 Просмотр списков

Поскольку в системе будет храниться множество однотипных данных, то должна существовать возможность просмотреть их список, каким-либо образом отфильтровать эти данные, либо найти какую-то конкретную запись. Для этого предназначены основные представления – индексы (Index).

На рисунке 10 представлена главная страница системы, отображающая список деталей, разбитый на страницы по 10 штук. Для удобства поиска необходимого компонента на форме присутствуют фильтры по категории и месту хранения и поиск по названию. Так выглядит эта страница для анонимного пользователя (по правому верхнему углу изображения видно, что пользователь не вошел в систему).

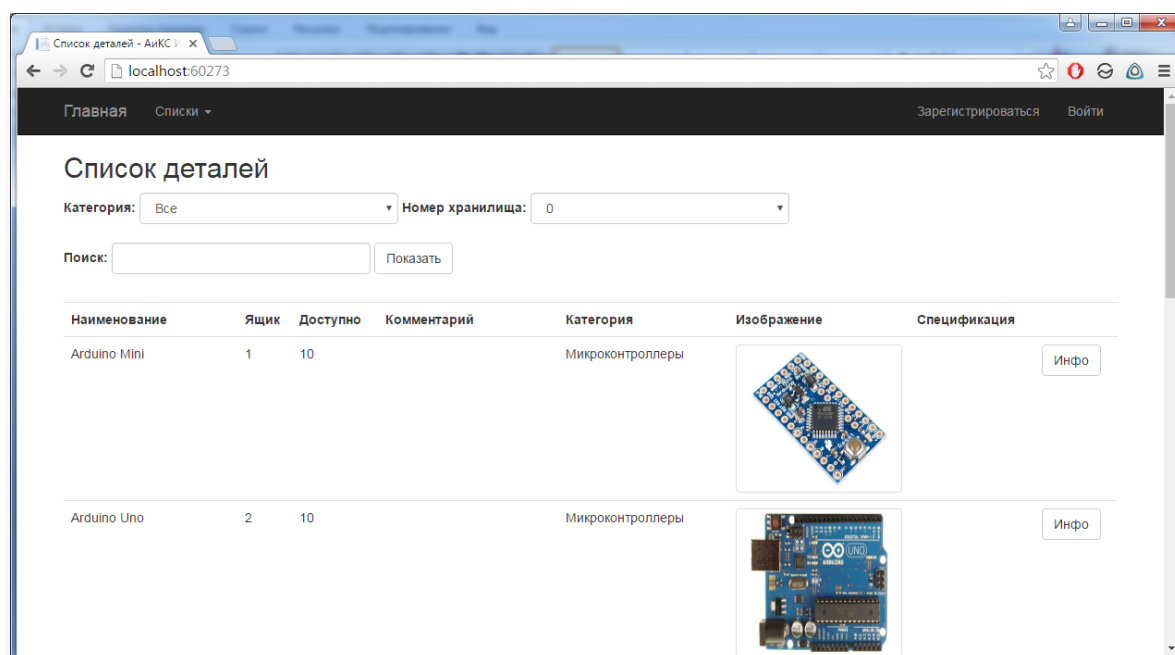


Рисунок 10 – Главная страница системы (список деталей)

В нижней части страницы расположены кнопки переключения между страницами. Внешний вид этих кнопок можно увидеть на рисунке 11. Чтобы не выводить один список деталей, который вполне может состоять и из тысячи строк, весь список выводится по 10 записей на одну страницу.

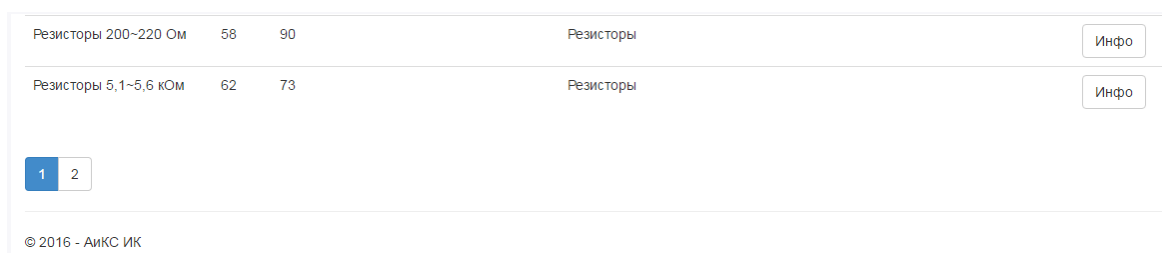


Рисунок 11 – Кнопки перехода между страницами

При авторизации[6] пользователя в системе, ему отобразится представление, чей внешний вид будет отличаться от представления, выводимого анонимному пользователю. Помимо кнопки «Инфо», которую можно видеть на предыдущих изображениях, появляется кнопка «Взять», после нажатия на которую пользователю будет выведено модальное окно, позволяющее выбрать необходимые параметры действия. Внешний вид модального окна показан на рисунке 12.

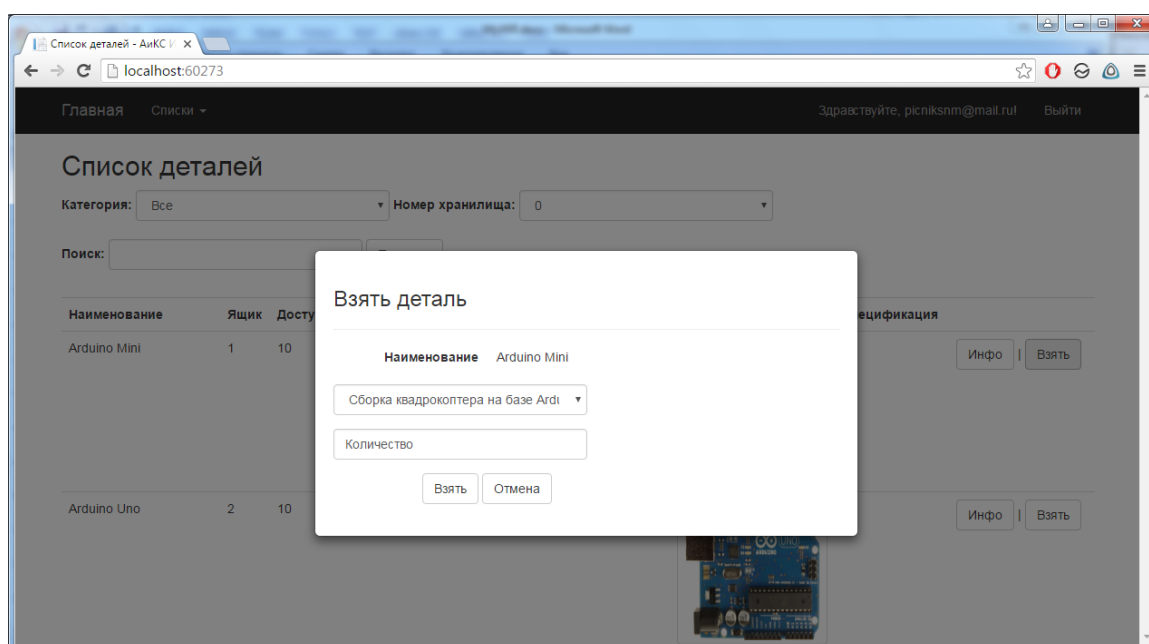


Рисунок 12 – Модальное окно «Взять деталь»

В выпадающем меню нужно выбрать один из проектов, в которых состоит пользователь, а в поле «Количество» необходимо указать, собственно, количество необходимых деталей.

Если в хранилище не достаточно необходимых деталей, можно посмотреть, в каких проектах такие детали задействованы, и, возможно, попро-

силь у студентов, занимающихся этим проектов, несколько недостающих деталей. Посмотреть подобную информацию можно в отдельном списке, который выводит все используемые на данный момент детали. На рисунке 13 изображен внешний вид этого списка.

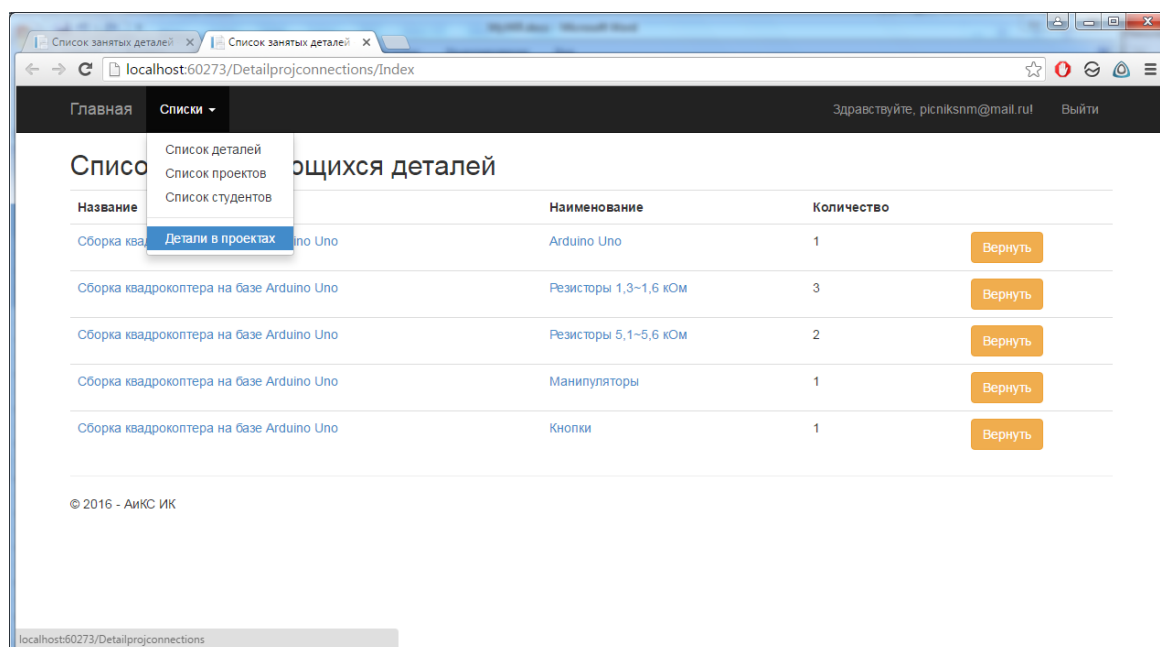


Рисунок 13 – Список задействованных в проектах деталей

Здесь же можно вернуть лишние детали из проектов, в которых состоит пользователь, нажав на кнопку вернуть. В этом случае появится модальное окно, в котором нужно будет указать количество лишних деталей.

Список проектов отображает все текущие проекты с возможностью фильтрации по статусу и поиска по названию. Здесь пользователь может присоединиться к проекту, если он в нем еще не состоит, или выйти из проекта, в котором он состоит. Студенты, состоящие в проекте, могут редактировать информацию о нем. Все это отображено на рисунке 14.

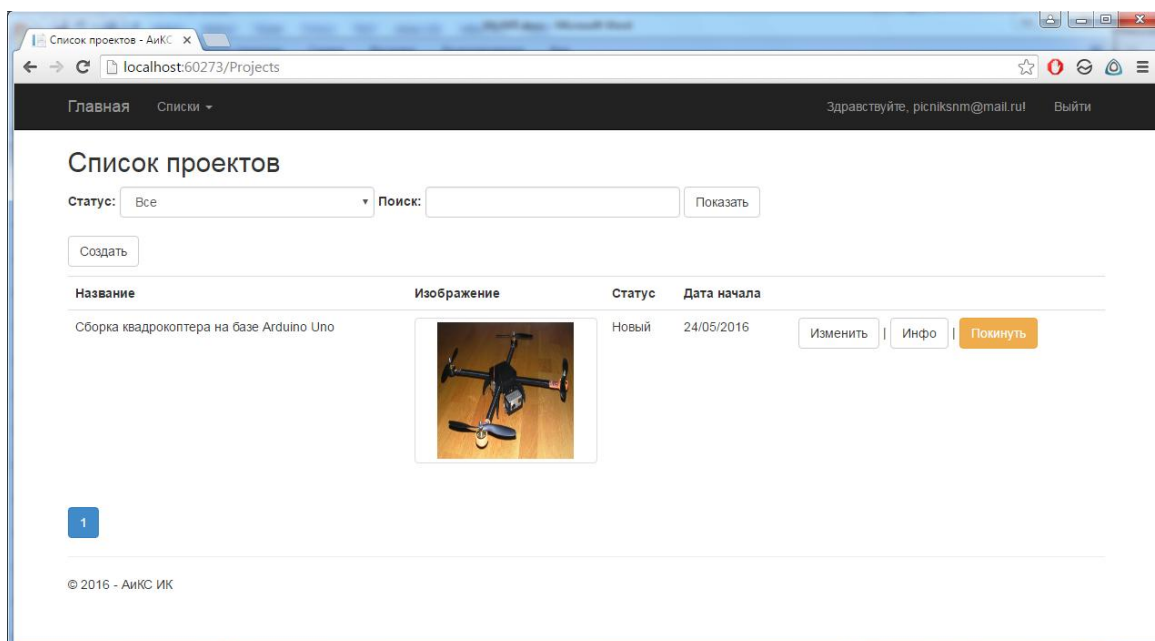


Рисунок 14 – Список проектов

Список пользователей отображает всех зарегистрированных в системе пользователей, за исключением администратора, и проекты, в которых они состоят. Здесь так же предусмотрена фильтрация по группе и поиск по фамилии пользователя. Страница со списком студентов изображена на рисунке 15.

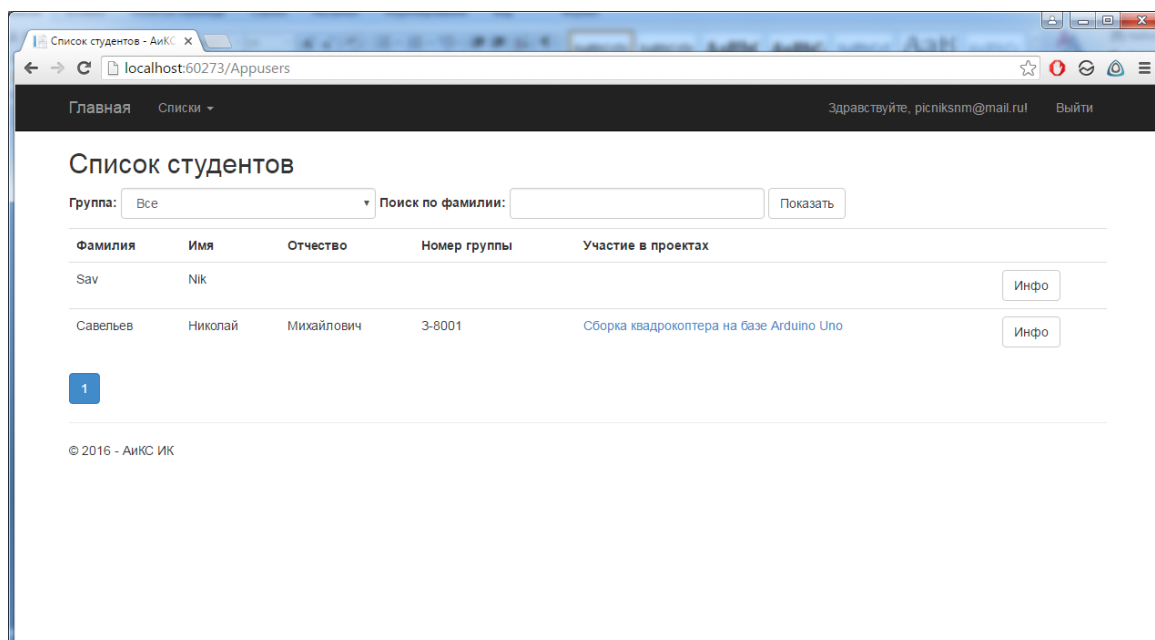


Рисунок 15 – Список зарегистрированных пользователей

В каждом из списков можно просмотреть более подробную информацию о любой записи, для этого служит кнопка «Инфо». Кроме того, все изо-

бражения можно увеличить для более детального просмотра, нажав на них. Полное изображение будет показано в модальном окне.

4.2 Возможности администратора

Для управления данными и контроля информационной системы предусмотрена учетная запись с администраторскими правами. Пользователь с данной ролью в системе обладает несколько отличными от обычных пользователей правами. Только администратор может создать новую запись в таблице деталей, изменить информацию о детали, списать или удалить ее. Это сделано для того, чтобы контроль над информацией, отражающей материальную базу лаборатории, оставался в руках ответственного за это человека. Изображение внешнего вида системы на странице списка деталей для администратора можно увидеть на рисунке 16.

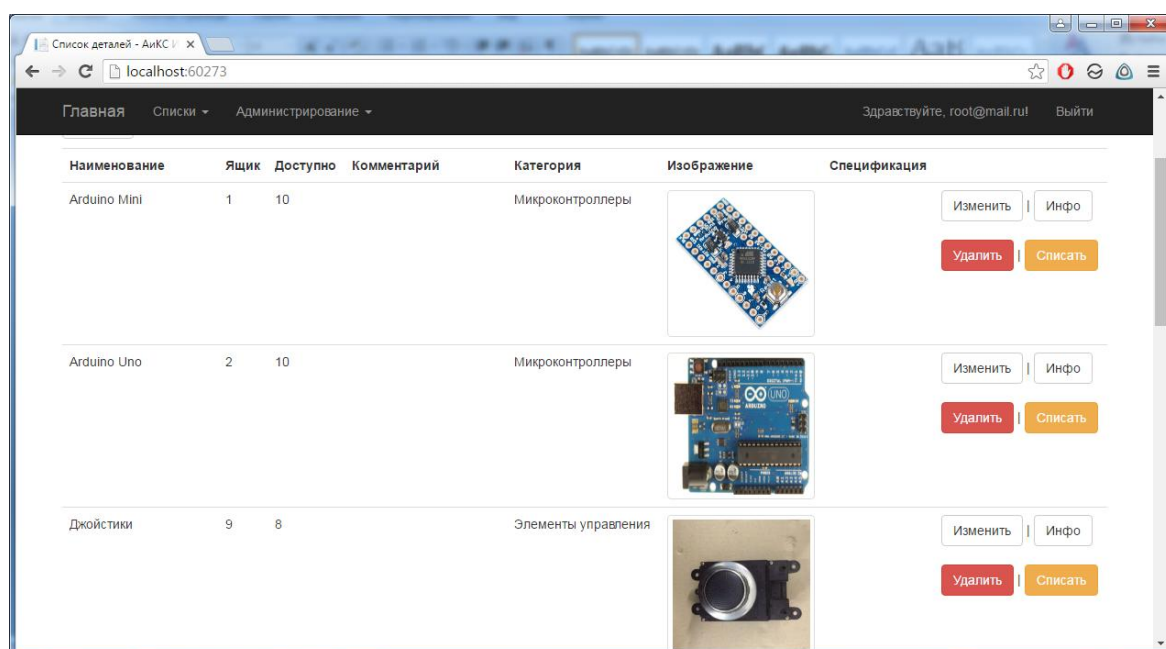


Рисунок 16 – Внешний вид списка деталей для администратора

Как можно заметить, после входа под учетной записью администратор, появляется новое выпадающее меню «Администрирование». В этом меню можно перейти к справочникам учебных групп, статусов проектов и категорий деталей. Отсюда же можно попасть к списку списанных деталей. Рабо-

та с этими справочниками разрешена только администратору. Только в справочник групп обычный пользователь может добавлять новые записи. Страницу справочника категорий и раскрытое меню «Администрирование» изображено на рисунке 17.

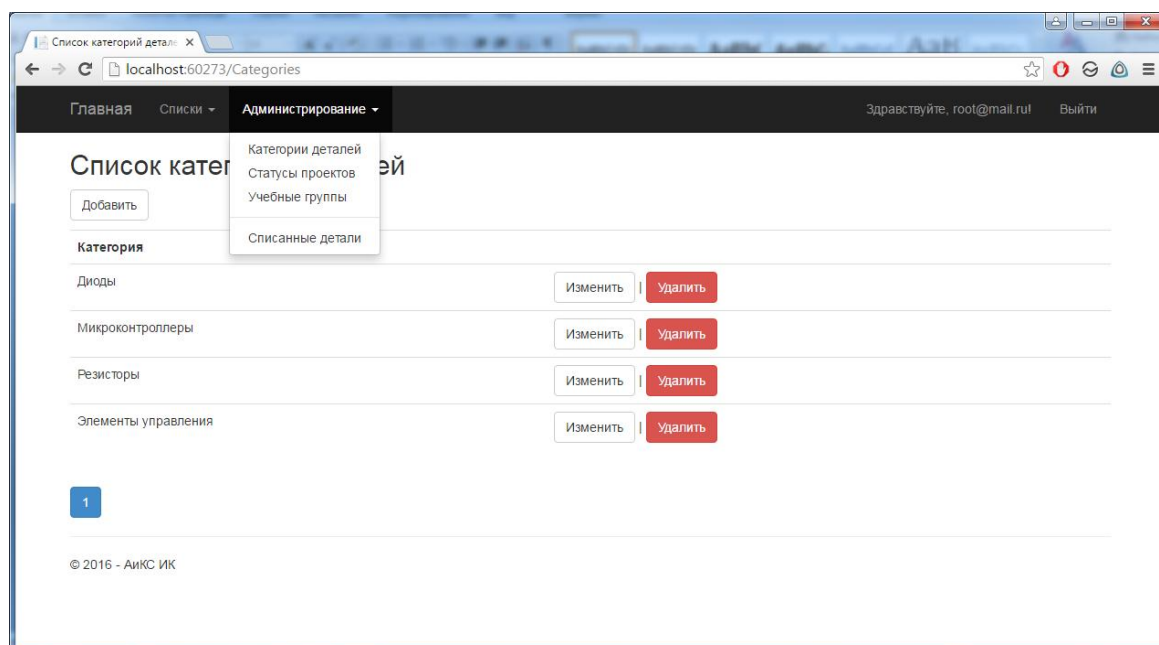


Рисунок 17 – Категории деталей, меню «Администрирование»

Рисунок 18 показывает процесс создания новой записи в таблице деталей.

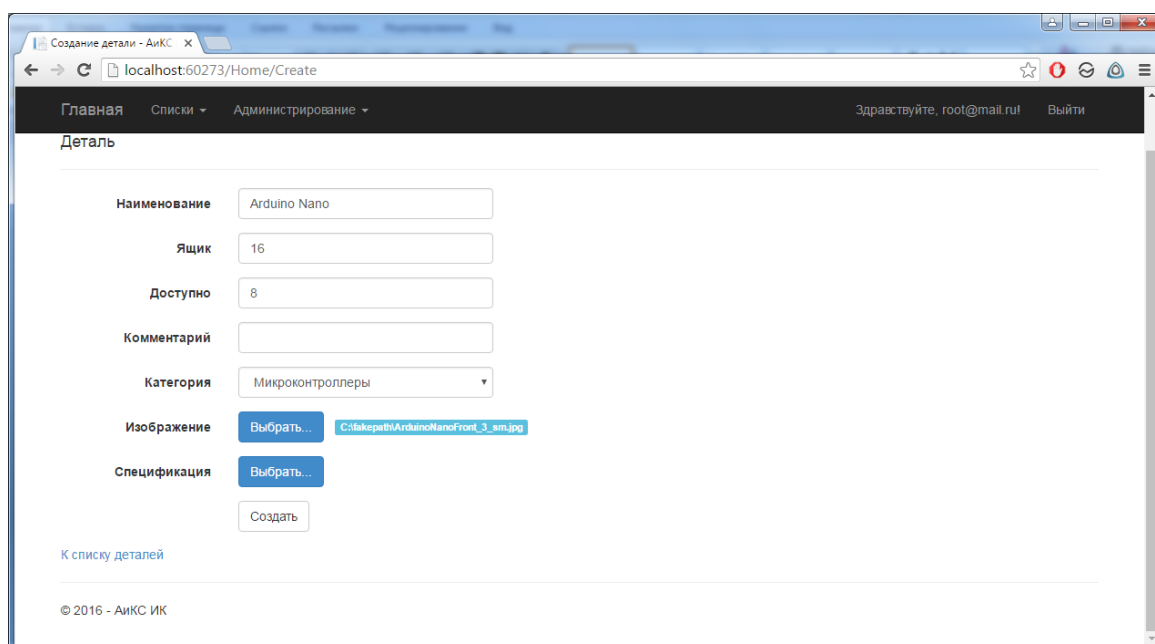


Рисунок 18 – Создание новой детали

Администратор может возвращать детали из любого проекта, но не может присоединяться к ним и, соответственно, брать детали. Администратор так же имеет возможность отредактировать любую запись из списка проектов. Кроме того, созданный пользователем проект может быть удален только администратором. Наличие этих возможностей можно увидеть на рисунке 19.

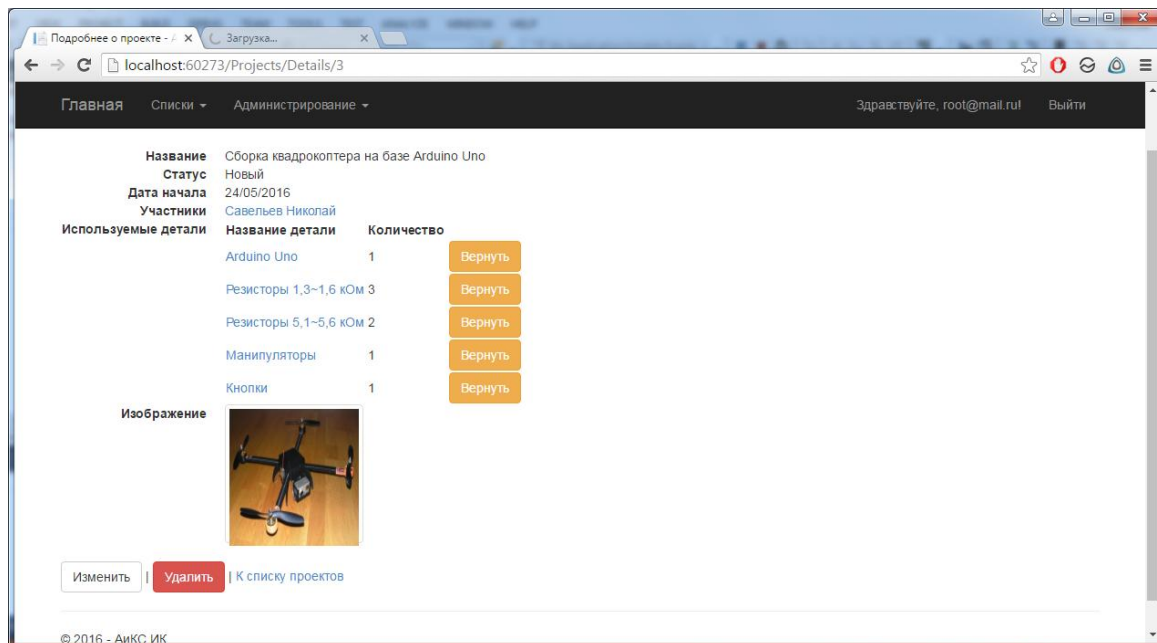


Рисунок 19 – Возможности администратора по управлению проектом

Таким образом, если сравнить возможности обычного пользователя и администратора, становится очевидно, что учетная запись администратора обладает гораздо большим набором инструментов по манипулированию данными нежели стандартные учетные записи пользователей. Следовательно, доступ к этой учетной записи должны иметь только сотрудники кафедры или ответственные за управление лабораторией люди, поскольку они будут полностью осознавать всю ответственность, которую возложит на них обладание полным контролем над практически всей информационной системой.

5 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

ТЭО необходимо для определения возможной величины экономического эффекта от использования результатов и наработок, полученных при исследовании и решении задач поставленных в дипломном проекте.

5.1 Обоснование необходимости и актуальности разработки

Подавляющее большинство бизнес процессов в современном мире имеют в своей основе некую материальную базу, используя которую они развиваются, приносят прибыль и создают рабочие места. Само понятие материальной базы очень широко и охватывает огромный круг всевозможных элементов, начиная с крупного рогатого скота в животноводстве и заканчивая авиалайнерами в гигантских авиакомпаниях.

Упорядоченное и эффективное использование материальной базы заметно повышает и эффективность всех процессов, которые, так или иначе, используют эту базу. Соответственно возрастает и экономическая составляющая эффективности данных процессов.

Для более полной отдачи от использования имеющейся в наличии материальной базы используются различные информационные системы, позволяющие систематизировать все данные обо всех элементах базы для более удобного и полного ее использования. Такие информационные системы отображают четкую картину использования ресурсов базы в любой произвольно взятый момент времени.

Данный проект имеет своей целью разработку такой информационной системы, направленной на оптимизацию использования ресурсов, используемых студентами в своих научных и исследовательских работах.

Поскольку такие системы являются узкоспециализированными под конкретную предметную область, то использование данной системы в других

областях и процессах станет возможно только после глубокой переделки самой системы для соответствия новой предметной области.

5.2 Организация и планирование комплекса работ

В данном проекте задействовано два человека – научный руководитель (сотрудник кафедры) и студент-дипломник, являющийся исполнителем проекта. Цель проекта выбрана совместными усилиями руководителя и исполнителя, техническое задание также разработано при участии обоих задействованных в проекте человек. Контроль над реализацией осуществляется научным руководителем вплоть до выпуска итоговой версии проектной системы. Реализация целиком возлагается на исполнителя.

Данная работа направлена на проектирование информационной системы, наиболее полно соответствующей предметной области, для которой разрабатывается, и позволяющей максимально оптимизировать использование наличествующей элементной базы.

Перечень всех необходимых работ и определение исполнителя для каждой из них происходит на этапе планирования. На этом же этапе происходит построение линейного графика работ, и устанавливается продолжительность каждого их этапов.

Итогом планирования стал план, состоящий из следующих этапов:

1. Подготовительный этап:
 - 1.1. Постановка целей и задач;
 - 1.1. Составление и утверждение технического задания;
 - 1.2. Подбор и изучение материалов по тематике;
 - 1.3. Обсуждение литературы;
2. Основной этап:
 - 2.1. Построение необходимых алгоритмов;
 - 2.2. Выбор программного обеспечения;

- 2.3. Написание программы, оформление записки;
- 2.4. Оформление материала
- 3. Заключительный этап:
 - 3.1. Подведение итогов

Наименование с указанной продолжительностью в процентном соотношении в отношении каждого из задействованных лиц указаны в таблице 2.

Таблица 2 – Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка задачи	НР	НР – 100%
Составление и утверждение технического задания (ТЗ)	НР, И	НР – 100% И – 10%
Подбор материала по направлению и тематике	НР, И	НР – 30% И – 100%
Разработка календарного плана	НР, И	НР – 100% И – 10%
Обсуждение литературы	НР, И	НР – 30% И – 100%
Построение алгоритмов	НР, И	НР – 100% ИП – 70%
Выбор программного обеспечения	НР, И	НР – 100% И – 80%
Написание программы (реализация)	И	И – 100%
Оформление материала в пояснительную записку	И	И – 100%
Представление графического материала	И	И – 100%
Представление результата	НР, И	НР – 60% И – 100%

5.2.1 Продолжительность этапов работ и календарный план

Существует два метода расчета этапа продолжительности работ:

а) технико-экономический метод;

б) опытно-статистический метод.

При отсутствии нормативной базы трудоемкости планируемых процессов использование технико-экономического метода становится невозможным, таким образом, остается только опытно-статистический метод, который может быть реализован двумя различными способами:

а) аналоговым;

б) экспертным.

Аналоговый способ возможен только при наличии соответствующего не устаревшего аналога проекта в целом, либо хотя бы некой его части (подразумевается соответствие значимых параметров).

При неимении же аналога, доступным остается лишь экспертный способ, который подразумевает генерацию необходимых оценок экспертами соответствующей предметной области на основе их профессионального опыта:

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (1)$$

где t_{min} – минимальная продолжительность работы, дн.;

t_{max} – максимальная продолжительность работы, дн.;

t_{prob} – наиболее вероятная продолжительность работы, дн.

Линейный график строится на основе рассчитанных длительностей этапов выполняемого проекта в рабочих днях, переведенных в календарные дни. Продолжительность выполнения одного конкретного этапа проекта ($T_{РД}$) рассчитывается по формуле:

$$T_{PD} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д} \quad (2)$$

где $t_{ож}$ – продолжительность работы, дн.;

$K_{ВН}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{ВН} = 1$;

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д} = 1-1,2$; в этих границах конкретное значение принимает сам исполнитель).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{PD} \cdot T_{К}, \quad (3)$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности, который позволяет перейти от длительности выполнения проекта в рабочих днях, к длительности выполнения проекта в календарных днях и наоборот:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (4)$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ} = 365$);

$T_{ВД}$ – выходные дни ($T_{ВД} = 52$);

$T_{\text{пд}}$ – праздничные дни ($T_{\text{пд}} = 10$).

$$T_{\text{к}} = \frac{365}{365 - 52 - 10} = 1,205$$

Таблица 3 отображает продолжительность этапов работ и их трудоемкость по занятым в них исполнителям. Столбцы 6 и 7 содержат величины трудоемкости этапа для каждого из двух участников проекта (научный руководитель и инженер) с учетом коэффициента $K_{\text{д}} = 1,2$. В столбцах 8 и 9 содержится информация о трудоемкости, выраженная в календарных днях. Величины трудоемкости этапов, рассчитанные по занятости в календарных днях по исполнителям $T_{\text{кд}}$, дают возможность построения линейного графика реализации проекта.

5.2.2 Календарный план-график работ

Календарный план-график работ представлен в виде линейного графика (диаграмма Ганта) в таблице 4, на котором работы по теме отображаются протяженными во времени отрезками.

Таблица 3 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка задачи	НР	3	5	3,8	3,8	0	4,579	0
Составление и утверждение технического задания (ТЗ)	НР, И	4	6	4,8	4,8	0,38	5,784	5,784
Подбор материала по направлению и тематике	НР, И	3	5	3,8	3,8	3,8	4,579	6,541429
Разработка календарного плана	НР, И	2	3	2,4	2,4	0,24	2,892	2,892
Обсуждение литературы	НР, И	2	4	2,8	2,8	3,8	3,374	5,623333
Построение алгоритмов	НР, И	6	10	7,6	7,6	5,32	9,158	10,17556
Выбор программного обеспечения	НР, И	3	4	3,4	3,4	2,72	4,097	4,097
Написание программы (реализация)	И	20	30	24	0	24	0	32,13333
Оформление материала в пояснительную записку	И	8	15	10,8	0	10,8	0	16,2675
Представление графического материала	И	7	10	8,2	0	8,2	0	10,97889
Представление результата	НР, И	6	8	6,8	6,8	6,8	8,194	9,104444
Итого:				78,4	35,4	85,97302	42,657	103,5975

Таблица 4

№ Работы	Наименование работы	t_{ki} , кал. дн.		Продолжительность выполнения работ, дни (клетка ~ 6 дней)																			
		НР	И	февраль					март					апрель					май				
1	Постановка задачи	4,58	-	■																			
2	Разработка и утверждение технического задания (ТЗ)	5,78	5,78		■																		
					■																		
3	Подбор материала по направлению и тематике	4,58	6,54			■																	
						■																	
4	Составление календарного плана	2,89	2,89				■																
							■																
5	Обсуждение литературы	3,37	5,62				■																
							■																
6	Выбор программного обеспечения	9,16	10,18				■	■															
							■	■															
7	Построение алгоритмов	4,1	4,1						■														
									■														
8	Расчет данных, написание программы	-	32,1							■	■	■	■	■	■	■	■	■	■	■	■	■	■
9	Оформление пояснительной записки	-	16,3												■	■	■	■	■	■	■	■	■
10	Представление графического материала	-	11																■	■	■	■	■
11	Представление результата	8,19	9,1																	■	■	■	■
																				■	■	■	■

5.2.3 Расчет накопления готовности проекта

Цель данного пункта – оценка текущих состояний (результатов) работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего (i -го) этапа выполнен общий объем работ по проекту в целом.

Введем обозначения:

- а) $TP_{\text{общ}}$ – общая трудоемкость проекта;
- б) TP_i (TP_k) – трудоемкость i -го (k -го) этапа проекта, $i = \overline{1, I}$;
- в) TP_{ij} (TP_{kj}) – трудоемкость работ, выполняемых j -м участником на i -м этапе, в нашем случае $j = \overline{1, m}$ – индекс исполнителя, поскольку исполнителя фактически двое, то $m = 2$.

Степень готовности определяется формулой:

$$CG_i = \frac{TP_i^H}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i TP_k}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}}. \quad (5)$$

Применительно к таблице 4 величины TP_{ij} (TP_{kj}) находятся в столбцах (8, $j = 1$) и (9, $j = 2$). $TP_{\text{общ.}}$ равна сумме чисел из итоговых клеток этих столбцов. Данные расчета степени готовности содержатся в таблице 5.

Таблица 5 – Нарастание технической готовности работы и удельный вес каждого этапа

Этап	TP_i , %	CG_i , %
Постановка задачи	3,13	3,12
Разработка и утверждение технического задания (ТЗ)	7,91	11,04
Подбор и изучение материалов по тематике	7,6	18,64
Разработка календарного плана	3,95	22,6
Обсуждение литературы	6,15	28,75
Выбор структурной схемы устройства	13,22	41,97
Выбор принципиальной схемы устройства	5,6	47,57
Расчет принципиальной схемы устройства	21,97	69,54
Оформление расчетно-пояснительной записки	11,12	80,67
Оформление графического материала	7,5	88,17

5.3 Расчет сметы затрат на разработку

В состав научно-исследовательской работы включаются затраты на все расходы, которые необходимы для выполнения утвержденного плана. Себестоимость НИР рассчитывается на основе планируемых затрат по приводимым статьям.

Для проведения данной НИР статьями расходов являются:

- а) покупные изделия и материалы;
- б) зарплата работникам;
- в) налоговые и прочие отчисления на социальные;
- г) оборудование для выполнения работ;
- д) услуги сторонних организаций;
- е) накладные расходы.

5.3.1 Расчет затрат на материалы

В данную статью входят: стоимость основных материалов, расходы на заказ, покупку и доставку. Стоимость материального имущества берется по каталожным розничным ценам на период разработки.

Транспортно-заготовительные расходы в проектно-расчетных ведомостях принимаются в размере 5-20% от отпускной цены закупаемых изделий и материалов.

Расчет затраты на материалы приведен в таблице 6.

Таблица 6 – Затраты на материалы и покупные изделия

Наименование материалов и покупных изделий	Единица измерения	Количество	Цена за ед., руб.	Сумма, руб
Бумага для ч/б печати А4	упаковка (20листов)	1	200	200
Картридж ч/б	шт.	1	1500	1500
персональные компьютеры	шт.	2	25000	50000
Принтер	шт.	1	4000	4000
Итого за материалы и покупные изделия:				55700
Транспортно-заготовительные расходы:				2785
Итого:				58485

Согласно таблице 6 расход на материалы составляет $C_{\text{мат}}=58485$ рублей.

5.3.2 Расчет по статье заработная плата

Данная статья расходов подразумевает основную заработную плату исполнителей (включая премии), непосредственно участвующих в разработке. Затраты потраченного времени на разработку для научного сотрудника и исполнителя в количестве рабочих дней и округленные до целого значения взяты с округлением до целого из таблицы 3. Премия за работу учитывается коэффициентом $K_{\text{ПР}} = 1,1$, дополнительная заработная плата учитывается коэффициентом $K_{\text{доп.ЗП}} = 1,188$, так же применяется районный коэффициент для Томска $K_{\text{р}} = 1,3$. Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку необходимо учесть интегральный коэффициент $K_{\text{и}} = 1,1 * 1,18 * 1,3 = 1,699$. Вышеуказанное значение $K_{\text{доп.ЗП}}$ применяется при шестидневной рабочей неделе, при пятидневной оно равно 1,113, соответственно в этом случае $K_{\text{и}} = 1,62$.

$$\text{ЗП}_{\text{дн-т}} = \text{МО} / 24,83, \quad (6)$$

где $ЗП_{\text{дн-т}}$ – среднедневная заработная плата.

В данной формуле учитывается, что в году 298 рабочих дней, в переводе на рабочие дни за месячный табель их получается 24,83.

Затраты труда, необходимые для выполнения i -го вида работ, в рабочих днях.

Среднедневная заработная плата определяется по формуле:

$$C_{\text{зн}_i} = \frac{D + D \cdot K}{F}, \quad (7)$$

где D - месячный должностной оклад работника, K - районный коэффициент (для Томска – 30%), F – количество рабочих дней в месяце (в среднем 22 дня).

Месячный должностной оклад исполнителя (специалист): $D = 6595,7$ рублей.

Месячный должностной оклад руководителя (ассистент): $D = 14584,32$ рублей.

Среднедневная заработная плата будет составлять по формуле (18):

$$C_{\text{зн.рук}} = \frac{14584,32 + 14584,32 * 0,3}{22} = 861,8 \text{ рублей}$$

$$C_{\text{зн.исп.}} = \frac{6595,7 + 6595,7 * 0,3}{22} = 389,75 \text{ рублей}$$

Найдем основную заработную плату и сведем полученные данные в таблицу 7.

Таблица 7

Участники разработки	Оклад, Руб./мес	Средняя ставка	Количество дней, дни	Коэф.	Основная зарплата Сосн. Зп, руб
Руководитель	14 584,32	861,8	42	1,699	61 496,32
Исполнитель	6595,7	389,75	103	1,62	65 033,68
Итого					126 530

5.3.3 Отчисления на социальные нужды

К отчислениям на социальные нужды относится так называемый социальный налог. По законодательству принятому в Российской Федерации на 01.01.2016 единый социальный налог составляет 30 %, поэтому:

$$C_{отч} = ECH \cdot (C_{осн/зп} + C_{доп/зп}), \quad (8)$$

$$C_{соц} = 126\,530 \cdot 0,30 = 37\,959 \text{ рублей.}$$

5.3.4 Расчет затрат на электроэнергию

К затратам на электроэнергию относятся расходы связанные с затратой электроэнергии, потраченной в ходе выполнения проекта.

$$C_{эл.об.} = P_{об} \cdot t_{об} \cdot Ц_{э}, \quad (9)$$

где $P_{об}$ – мощность, потребляемая оборудованием, кВт;

$Ц_{э}$ – тариф на 1 кВт·час;

$t_{об}$ – время работы оборудования, час.

Для ТПУ $Ц_{э} = 5,257 \text{ руб./кВт·час (с НДС).}$

На основе данных сведенных в таблицу №5.2 время работы оборудования можно принять равным времени затраченного НР и И при разработке проекта:

$$t_{об} = T_{рД} * K_t, \quad (10)$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{рД}$, определяется исполнителем самостоятельно.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном.} * K_C, \quad (11)$$

где $P_{ном.}$ – номинальная мощность оборудования, кВт;

$K_C \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_C = 1$.

Затраты на электроэнергию приведены в таблице 8.

Таблица 8

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $\mathcal{E}_{об}$, руб.
Персональный компьютер	718,21	0,3	1132,69
Струйный принтер	35	0,1	18,4
Итого:			1151,1

5.3.5 Расходы на оборудование для выполнения работ

Расчет расходов на оборудование для выполнения работ приведен в пункте 4.3.1.

5.3.6 Расчет амортизационных расходов

В этой статье учитываются расходы, связанные с эксплуатацией персонального компьютера и принтера.

Стоимость машинного времени, потраченного на проектирование, рассчитывается как:

$$C_{AM} = \frac{H_A * Ц_{ОБ} * t_{рф} * n}{F_D}, \quad (12)$$

где H_A – амортизация годовой нормы единицы оборудования;

$Ц_{ОБ}$ – балансовая стоимость единицы оборудования с учетом ТЗР;

F_D – действительный годовой фонд времени работы соответствующего оборудования, берется из специальных справочников или фактического режима его использования в текущем календарном году. При этом второй вариант позволяет получить более объективную оценку C_{AM} ;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

n – число задействованных однотипных единиц оборудования.

Из документа Российской Федерации «О классификации основных средств, включенных в амортизационные группы». Можно рассчитать амортизацию годовой нормы. Например, для ПК это $2 \div 3$ года. Необходимо задать конкретное значение $С_A$, например, 2,5 года. Далее определяется H_A , как величина, обратная $С_A$, в данном случае это $1 : 2,5 = 0,4$. Расчет приведен в таблице 9.

Таблица 9

Наименование оборудования	Годовая норма амортизации H_A	Балансовая стоимость $Ц_{ОБ}$, руб	Фактическое время работы $t_{рф}$, ч	$C_{ам}$, руб
Персональный компьютер	0,4	25000	722	6057
Струйный принтер	0,2	4000	30	48
Итого:				6105

5.3.7 Расходы на услуги сторонних организаций

В этой статье учитываются расходы, связанные с полученными в процессе проектирования услугами от сторонних организаций. Перечень и их стоимость приведена в таблице 10.

Таблица 10 – Затраты на услуги сторонних организаций

Услуга	Количество	Стоимость одной единицы, руб.	Сумма затрат, руб.
Переплёт	4	30,0	120,0
Итого:			120,0

5.3.8 Прочие расходы

К прочим расходам отнесены расходы неучтенные в работе ранее, они принимаются равными 10% всей суммы.

$$C_{\text{проч}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}} + C_{\text{нп}}) * 0,1 = 22905,31$$

5.3.9 Расчет общей себестоимости

Смета затрат на разработку проекта приведена в таблице 11.

Таблица 11

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	58485
Основная заработная плата	$C_{\text{зп}}$	126 530
Отчисления в социальные фонды	$C_{\text{соц}}$	36782
Расходы на электроэнергию	$C_{\text{эл.}}$	1151,1
Амортизационные отчисления	$C_{\text{ам}}$	6105
Прочие расходы	$C_{\text{проч}}$	22905,31
Итого:		251 958,41

Таким образом, затраты на разработку составили $C = 251\,958,41$ руб.

5.3.10 Расчет прибыли

Прибыль от реализации проекта в зависимости от конкретной ситуации (масштаб и характер получаемого результата, степень его определенности и коммерциализации, специфика целевого сегмента рынка и т.д.) может определяться различными способами. Если исполнитель работы не располагает дан-

ными для применения «сложных» методов, то прибыль следует принять в размере $5 \div 20 \%$ от полной себестоимости проекта. В нашем случае она составляет 25195,841 руб. (10 %) от расходов на разработку проекта.

5.3.11 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. В нашем случае это $(251\,958,41 + 25195,841) * 0,18 = 49887,76$ руб.

5.3.12 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае:

$$C_{\text{НИР(КР)}} = 251\,958,41 + 25195,841 + 49887,76 = 321439,9 \text{ руб.}$$

5.4 Оценка экономической эффективности проекта

Поскольку применять данную информационную систему предполагается в процессе обучения, следовательно, прямой прибыли данная разработка не принесет. Однако данная система повлияет непосредственно на сам процесс обучения. Студенты, реализующие свои собственные научные проекты на базе лаборатории студенческого научного творчества, получают возможность более качественного планирования дальнейшего развития своего проекта. Как следствие, сократится среднее время, потребное на завершение одного проекта. Кроме того, при правильном планировании возрастет и качество студенческих разработок.

Помимо студентов, разработка поможет кафедре в учете деталей, которые закупаются на деньги кафедры. Это поможет уменьшить шанс утери некоторых деталей, принадлежащих лаборатории. Еще одним плюсом внедрения данной системы будет значительное упрощение процесса планирования новых закупок для нужд лаборатории.

5.5 Оценка научного уровня

Для оценки научно-технического эффекта данной разработки будет использоваться метод бальных оценок.

Метод заключается в том, что разработка оценивается по нескольким показателям, каждому из которых по принятой шкале присваивается определенное количество баллов.

На основе оценок новизны результатов, их ценности, масштабам реализации определяется показатель научно-технического уровня по формуле:

$$H_m = \sum_{i=1}^n K_i \cdot P_i, (13)$$

где K_i - весовой коэффициент i -го признака научно-технического эффекта;

P_i - количественная оценка i -го признака научно-технического уровня работы.

Приблизительные коэффициенты приведены в таблице 12.

Таблица 12 – Признаки научно-технического эффекта

Признак научно-технического эффекта НИ-ОКР(i)	Примерные значения весового коэффициента (K_i)
Уровень новизны	0,6
Теоретический уровень	0,4
Возможные реализации	0,2

Количественная оценка уровня новизны НИР определяется на основе значения баллов по таблице 13.

Таблица 13 – Количественная оценка уровня новизны НИР

Уровень новизны разработки	Характеристика уровня новизны	Баллы
Принципиально новая	Результаты исследований открывают новое направление в данной области науки и техники	8 - 10
Новая	По-новому или впервые объяснены известные факты, закономерности	5 - 7
Относительно новая	Результаты исследований систематизируют и обобщают имеющиеся сведения, определяют пути дальнейших исследований	2 - 4
Традиционная	Работа выполнена по традиционной методике, результаты которой носят информационный характер	1
Не обладающая новизной	Получен результат, который ранее был известен	0

Теоретический уровень полученных результатов НИР определяется на основе значения баллов, приведенных в таблице 14.

Таблица 14 – Количественная оценка теоретического уровня НИР

Теоретический уровень полученных результатов	Баллы
Установление закона; разработка новой теории	10
Глубокая разработка проблемы: многоаспектный анализ связей, взаимозависимости между фактами с наличием объяснения	8
Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	6
Элементарный анализ связей между фактами с наличием гипотезы, симплексного прогноза, классификации, объясняющей версии или практических рекомендаций частного характера	2
Описание отдельных элементарных фактов (вещей, свойств и отношений); изложение опыта, наблюдений, результатов измерений	0,5

Возможность реализации научных результатов определяется на основе значения баллов по таблице 15.

Таблица 15 – Возможность реализации научных результатов

Время реализации	Баллы
В течении первых лет	10
От 5 до 10 лет	4
Более 10 лет	2
Масштабы реализации	Баллы
Одно или несколько предприятий	2
Отрасль (министерство)	4
Народное хозяйство	10

Определяем показатель научно-технического эффекта, который оценивается в зависимости от результата.

Итоговая количественная оценка признаков НИР приведена в таблице 16.

Таблица 16 – Количественная оценка признаков НИР

Признак научно-технического эффекта НИР	Характеристика признака НИР	K_i	P_i
1. Уровень новизны	Систематизируют и обобщают сведения, определяют пути дальнейших исследований	0,6	4
2. Теоретический уровень	Глубокая разработка проблемы: многоаспектный анализ связей, взаимозависимости между фактами с наличием объяснения	0,4	8
3. Возможность реализации	Время реализации в течение первых лет	0,2	10
	Масштабы реализации - предприятие	0,2	2

Результаты оценок признаков отображены в таблице 17.

Таблица 17 – Оценка уровня научно-технического эффекта

Уровень научно-технического эффекта	Показатель научно-технического эффекта
Низкий	1 - 4
Средний	5 - 7
Сравнительно высокий	8 - 10
Высокий	11 - 14

Показатель научно технического эффекта равен:

$$H = 0,6 \cdot 4 + 0,4 \cdot 8 + 0,2 \cdot 10 + 0,2 \cdot 2 = 8$$

В соответствии с таблицей 14 уровень научно-технического эффекта данной разработки – сравнительно высокий.

6 Социальная ответственность

Раздел социальной ответственности предназначен для рассмотрения различных вероятных вредоносных факторов, которые могут возникнуть в случае сбоев в работе разрабатываемой в данном проекте системы. Здесь же будут описаны различные мероприятия, которые были приняты для устранения возможных причин сбоев, и рекомендации по предотвращению тех причин, возникновение которых не может быть предусмотрено в работе самой системы.

Поскольку данная система предназначена в первую очередь для учета определенной материальной базы, то прямой возможности возникновения чрезвычайных происшествий не существует. Однако это не исключает возможности возникновения самих сбоев в работе системы.

6.1 Анализ возможных сбоев и их последствий

Информационная система учета элементной базы лаборатории студенческого научного творчества предназначена для систематизации информации о студентах, ведущихся ими проектах и используемых в них различных электронных, электрических и механических деталях. Поскольку система не является автоматизированной, то подразумевается, что внесение новых и изменение уже внесенных данных будут выполняться пользователями системы, что автоматически подразумевает возможность занесения ошибочных данных. Так как система не предназначена для использования в промышленных производствах (в том числе химических, фармацевтических и т.д.), то и последствия такой ошибки могут быть значительно менее опасными. Однако даже в таком случае нельзя совсем игнорировать возможность возникновения таких происшествий.

В первую очередь, сбой возможен при проблемах с соединением между компьютерами. Поскольку данная система разрабатывается как клиент-серверное приложение, то все данные, находящиеся в системе, хранятся имен-

но на сервере. В случае данного сбоя станет невозможна работа с системой на любых компьютерах, за исключением самого сервера.

В данном случае угрозы здоровью и материальному состоянию людей, находящихся в момент сбоя в непосредственной близости от компьютеров, подключенных к системе, не возникает. Данные же могут быть потеряны только те, которые не успели отправить на сервер до потери соединения с сервером.

Далее, существует возможность нештатного отключения сервера. В случае возникновения данной ситуации, возможные последствия практически идентичны предыдущему случаю, за исключением возможности повреждения данных, хранящихся на сервере. Это может произойти вследствие выхода из строя носителя информации, на котором располагается база данных, с которой работает система.

Кроме того, возможно возникновение ситуации, когда пользователь случайно, либо целенаправленно произведет действия, могущие привести к сбою в системе. В процессе выполнения такого действия возникнет ошибка, которая будет выведена пользователю, а само действие будет прекращено. Работа же системы прекращена не будет.

И наконец, возможен момент, когда пользователь, опять же случайно или целенаправленно, допустит несоответствие между данными, хранящимися в базе данных системы и реальными данными. В таком случае, круг возможных последствий значительно расширяется. Например, если неисправная деталь в системе будет отображаться как исправная, то она вполне может быть использована студентами в своем проекте. Соответственно, весь проект может оказаться не рабочим. И даже более того, остальные детали могут так же выйти из строя из-за неисправности лишь одной детали. Это означает дополнительные денежные траты для кафедры ради замещения вышедших из строя деталей и безвозвратно потерянное студентами время.

Однако проект вполне может получиться рабочим, что может оказаться даже более опасным, ведь в процессе работы такого проекта могут происходить сбои из-за неисправной детали. В результате таких сбоев могут быть повреждены окружающие предметы, а люди, находящиеся в данный момент рядом, могут получить различные травмы от движущихся деталей нестабильно работающего проекта.

Поскольку в проектах помимо электронных и механических используются также электрические компоненты, то это подразумевает работу с электрическим током. В случае же использования неисправной детали при работе с электрической сетью, возможно возникновение короткого замыкания, которое может привести к пожару, который, в свою очередь, несет значительную угрозу здоровью и жизни людей. Даже при отсутствии возгорания, существует вероятность удара электрическим током, который может быть крайне опасен для человека.

6.2 Встроенная защита системы от сбоев

Для того чтобы нивелировать возможные последствия различных сбоев в работе системы, необходимо еще на этапе проектирования предусмотреть максимальное количество факторов, влияющих на возникновение этих сбоев, и постараться защитить систему от их влияния.

В самой системе встроена защита от некоторых очевидных моментов, могущих привести к сбоям в работе всей системы.

Так, для отсечения ввода заведомо некорректных данных используется валидация. Она обеспечивает проверку вносимых в систему данных на соответствие требованиям к этим данным. Например, в поле для ввода количества должно быть внесено неотрицательное числовое значение, любые другие данные не пройдут валидацию и пользователю будет выведена ошибка.

Примера кода валидации для поля ввода количества:

```
[Required(ErrorMessage = "Поле не должно быть пустым")]  
  
[Range(0, int.MaxValue, ErrorMessage = "Количество не может быть отри-  
цательным")] ]
```

Однако данный метод защиты предотвращает ввод только очевидно неподходящих данных и не обеспечивает полной защиты от ввода ошибочной информации. Так, например, в то же самое поле для ввода количества вполне может быть занесено числовое значение, которое не соответствует реальному количеству.

Для защиты системы от сбоев при некоторых некорректных действиях пользователя, как, например, удаление данных, связанных с другими данными в системе, в коде присутствуют необходимые проверки, которые не позволят произойти сбою. От случайного же удаления данные защищены дополнительной проверкой, которая требует подтверждения действия удаления.

Пример кода проверки на связанность данных при удалении Группы из справочника:

```
var user = db.Users.Where(p => p.GroupId == id).FirstOrDefault();  
Group group = db.Groups.Find(id);  
if (user == null)  
{  
    db.Groups.Remove(group);  
    db.SaveChanges();  
    return RedirectToAction("Index");  
}  
else  
{  
    return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
}
```

В случае, если не существует ни одного пользователя, которому присвоено значение данной Группы, то происходит удаление этой Группы, иначе – выводится сообщение об ошибке.

6.3 Мероприятия по защите оборудования

Поскольку разрабатываемый проект представляет собой чисто программный комплекс, то логично, что воздействие факторов, влияющих на физическую платформу, на базе которой будет разворачиваться система, невозможно предотвратить средствами самой системы. Для защиты от таких факторов необходимо обеспечить платформу физическими средствами, которые позволяют игнорировать данные факторы.

Устранить угрозу сбоев в питании достаточно просто – установка источника бесперебойного питания (ИБП) создаст необходимый запас времени для штатного выключения платформы при внезапном отключении электричества. Так же ИБП предоставит защиту платформы от скачков напряжения в сети. Типичный внешний вид ИБП представлен ниже на рисунке 20.



Рисунок 20 – Внешний вид источника бесперебойного питания

Обезопасить соединение между компьютерами для доступности системы с других рабочих мест помимо самого сервера несколько сложнее. Повреждение кабеля возможно предотвратить путем его прокладки в специальный

кабель-канал. Это не гарантирует его 100%-ой защищенности, однако значительно снижает вероятность данного происшествия.

Сетевое оборудование, обеспечивающее соединение между компьютерами, на которых используется система, и сервером, на котором система непосредственно расположена, так же должно быть подключено к ИБП для того, чтобы предоставить возможность закончить передачу данных на сервер и не потерять их.

6.4 Мероприятия по защите достоверности данных

Поскольку полностью предотвратить ввод ошибочных данных программными средствами невозможно, так как здесь присутствует человеческий фактор, то для управления системой и контроля достоверности данных необходимо назначение администратора, который будет нести ответственность в случае возникновения различных последствий, возникших из-за несоответствия данных, занесенных в систему, и реальных данных.

Администратор и обычные пользователи имеют различные права и возможности в системе, что подразумевает собой и различные степени ответственности за те или иные действия. Так, например, администратор имеет полный доступ к справочникам, хранящимся в базе данных, и только администратор может управлять записями, описывающими детали (добавление, редактирование, удаление) и т.д.

Для такого разграничения разрешений на выполнение операций в коде системы предусмотрены множественные проверки наличия у текущего пользователя, вошедшего в систему, роли администратора.

Ниже приведена таблица, отображающая разграничение прав доступа к различным операциям, заложенным в системе (таблица 18).

Таблица 18 – Разграничение прав доступа

Список	Действие	Администратор	Пользователь	Аноним
Детали	Списать	+	-	-
	Взять	-	+ (Для проектов, в которых участвует)	-
	Вернуть	+	+ (Для проектов, в которых участвует, в которых есть детали)	-
	Просмотр	+	+	+
	Просмотр подробностей	+	+	+
	Добавить	-	+	-
	Изменить	+	-	-
	Удалить	+	-	-
Списанные детали	Восстановить	+	-	-
	Удалить	+	-	-
Проекты	Участвовать	-	+ (Если еще в нем не участвует)	-
	Покинуть	-	+ (Если уже в нем участвует)	-
	Просмотр	+	+	+
	Просмотр подробностей	+	+	+
	Добавить	-	+	-
	Изменить	+	+ (Только в котором участвует)	-
	Удалить	+		-

Продолжение таблицы 18

Список	Действие	Администратор	Пользователь	Аноним
Пользователи	Просмотр	+	+	+
	Просмотр подробностей	+	+	+
	Изменить	+	+(Только свою запись)	-
	Удалить	+	-	-
Статусы проектов	Просмотр	+	-	-
	Добавить	+	-	-
	Изменить	+	-	-
	Удалить	+	-	-
Номера групп	Просмотр	+	-	-
	Добавить	+	-	-
	Изменить	+	-	-
	Удалить	+	-	-
Категории деталей	Просмотр	+	-	-
	Добавить	+	-	-
	Изменить	+	-	-
	Удалить	+	-	-

Администратор должен четко осознавать свои обязанности и надлежащим образом выполнять их.

От человека на должности администратора потребуется исполнительность и ответственность для регулярных проверок актуальности данных и внимательность для нахождения ошибок во внесенной в систему информации.

Кроме того, на администратора возлагается обязанность по отслеживанию количества свободных деталей различных типов. Отсутствие своевременной реакции на недостаток определенных деталей может привести к задержкам в развитии некоторых текущих проектов, для которых эти детали могут понадобиться.

Невнимательность администратора системы так же может привести и к безвозвратной утере каких-либо элементов, что повлечет за собой лишние расходы на приобретение замены для утерянной детали.

Пользователей системы необходимо ознакомить с правилами пользования информационной системой для снижения рисков возникновения несоответствия данных, однако это не исключит их полностью.

Для совершения необходимых действий, пользователь должен будет зарегистрироваться и войти в систему, анонимные пользователи смогут только просмотреть списки проектов, деталей и зарегистрировавшихся пользователей. Изменять, удалять, добавлять данные в систему они не смогут.

Таким образом, ответственность за корректность работы информационной системы целиком возлагается на администратора, назначенного контролировать эту систему. Именно он должен будет следить за актуальностью вносимых и внесенных данных и проверять все изменения в них.

ЗАКЛЮЧЕНИЕ

В процессе наблюдения за работой лаборатории были выявлены узкие места в процессе обработке сотрудниками лаборатории хранящихся в ней данных. В контексте данной работы было проведено глубокое исследование деятельности лаборатории студенческого научного творчества и протекающих внутри процессов для нахождения возможных способов оптимизации узких мест таким образом, чтобы это позволило ускорить и повысить качество ее работы в целом. Результатом этого исследования стало принятие решения о целесообразности разработки информационной системы учета для лаборатории.

Для выполнения этой работы была изучена номенклатура предметной области лаборатории и необходимые для проведения разработки технологии, построена модель, описывающая номенклатурные элементы лаборатории и их взаимосвязи.

На базе полученной модели и с применением изученных технологий была разработана информационная система учета элементной базы лаборатории студенческого научного творчества. Данная система позволяет оптимизировать работу сотрудников лаборатории с записями хранения, выдачи и получения обратно элементов, входящих в перечень элементной базы лаборатории. Кроме того, система позволяет студентам-участникам лаборатории изучить все наличествующие компоненты и максимально эффективно спланировать процесс развития своего исследования. Система так же позволяет частично переложить обязанность по ведению учета на студентов.

При ответственном наполнении данными система позволит следить за процессом развития как студенческих проектов, так и лаборатории в целом, позволяя своевременно корректировать курс ее развития.

CONCLUSION

Bottlenecks have been found in the process of laboratory staff processing the data stored in it while observing the work of the laboratory. In the context of this work study of the laboratory of the student scientific work and the processes occurring inside was conducted to find possible ways to optimize the bottlenecks so that it is possible to speed up and improve the quality of its work as a whole. The result of this research was the adoption of a decision on the feasibility of developing an information accounting system for the laboratory.

The nomenclature of the domain of the lab and all necessary technologies of development were studied to carry out this work. As a result a model which describes the elements of the nomenclature and their relationships was built.

On the basis of the received model and using the studied technologies an information system for recording elemental base of the laboratory of the student scientific work was developed. This system allows to optimize the performance of the laboratory staff while processing different records such as storing, issuing and receiving back the elements included into the list of components of the laboratory's elemental base. The system also allows students who participate in the laboratory's projects to examine all components and to plan the development of their research as efficiently as possible. The system also allows to shift the responsibility for record-keeping on students partially.

If the system filled with data responsibly then it could allow user to monitor the process of development of the student projects and laboratory as a whole, allowing timely adjustments to the course of its development.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Руководство по ASP .NET MVC 5 [Электронный ресурс]
URL: <http://metanit.com/sharp/mvc5/>, свободный. - Загл. с экрана. - Яз. рус.
2. Адам Фримен. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов, 5-е издание. — М.: «Вильямс», 2014. — 736 с.
3. Джозеф Албахари, Бен Албахари. C# 5.0. Справочник. Полное описание языка. — М.: «Вильямс», 2013. — 1008 с.
4. Адам Фримен. ASP.NET 4.5 с примерами на C# 5.0 для профессионалов, 5-е издание. — М.: «Вильямс», 2014. — 1120 с.
5. Руководство по Entity Framework [Электронный ресурс]
URL: <http://metanit.com/sharp/entityframework/>, свободный. - Загл. с экрана. - Яз. рус.
6. ASP.NET 5 Identity 3 и новый инструментарий аутентификации/авторизации [Электронный ресурс]
URL: <https://habrahabr.ru/company/nixsolutions/blog/273753/>, свободный. - Загл. с экрана. - Яз. рус.
7. Кристиан Нейгел и др. C# 5.0 и платформа .NET 4.5 для профессионалов. — М.: «Диалектика», 2013. — 1440 с.
8. Кристиан Нейгел, Карли Уотсон и др. Visual C# 2010: полный курс. — М.: Диалектика, 2010.
9. Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4.0. — М.: «Вильямс», 2010. — С. 1392.
10. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд. Язык программирования C#. Классика Computers Science. 4-е издание. — СПб.: «Питер», 2012. — 784 с.